

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Android pro handicapované**

# **Prohlášení**

Prohlašuji, že diplomovou práci jsem vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne

## **Abstract**

The main purpose of this thesis is development of an application for people with disabilities. The first part consists of an analysis, where the individual disabilities are divided. The main functionalities of the application have been determined. In the following part there is described implementation and the classes and their methods are defined. In the testing section there are clarified the main problems, which occurred while the application was tested. The conclusion consists of description of the results and some of the possible improvements are mentioned.

## **Abstrakt**

Práce se zabývá vývojem aplikace pro osoby se zdravotním postižením. V první části je uvedena analýza sestávající z kategorizace jednotlivých postižení, ke kterým je přiřazeno několik již dostupných aplikací. Dále byly stanoveny hlavní funkcionality a cíle práce. Následně je nastíněna implementace aplikací a jsou popsány třídy a jejich metody. V části testování jsou popsány problémy, které se během této fáze vyskytly. V závěru práce jsou zmíněna možná rozšíření aplikace.

## **Poděkování**

Na tomto místě bych rád poděkoval vedoucímu své diplomové práce Ing. Ladislavu Pešíčkovi za odborné vedení a cenné rady, které mi pomohly tuto práci úspěšně dokončit.

# Obsah

1.	Úvod.....	1
2.	Analýza.....	3
2.1.	Asistivní technologie .....	3
2.2.	Kategorizace postižení.....	4
2.2.1.	Zrakové postižení.....	5
2.2.2.	Sluchové postižení .....	5
2.2.3.	Tělesné postižení.....	6
2.2.4.	Mentální postižení.....	6
2.2.5.	Vnitřní postižení .....	8
2.2.6.	Duševní nemoci .....	9
2.2.7.	Kombinovaná postižení .....	10
2.2.8.	Senioři.....	10
3.	Možnosti Androidu .....	11
3.1.	Podpora v Androidu.....	11
3.2.	Zrakové postižení .....	12
3.2.1.	Braille Helper.....	13
3.2.2.	ColorBlindness SimulateCorrect .....	13
3.2.3.	Be My Eyes.....	13
3.2.4.	Learn Braille Alphabet.....	13
3.3.	Sluchové postižení.....	14
3.3.1.	RogerVoice .....	14
3.3.2.	The Deaf And Hearing Impaired .....	15
3.3.3.	Deaf - Hearing chat device .....	15
3.3.4.	Spread Signs .....	15
3.4.	Tělesné postižení .....	16

3.4.1.	Commandr for Google Now .....	17
3.4.2.	Tecla Access .....	17
3.5.	Mentální postižení .....	17
3.5.1.	Avaz - AAC App for Autism.....	18
3.5.2.	Talkitt.....	18
3.6.	Vnitřní postižení .....	18
3.6.1.	MySugr Diabetes Logbook.....	19
3.6.2.	My Epilepsy Diary.....	19
3.6.3.	Find Me Gluten Free.....	19
3.6.4.	ZYRTEC® AllergyCast .....	19
3.7.	Duševní nemoci .....	20
3.7.1.	MoodTools - Depression Aid .....	20
3.7.2.	Code Blue .....	22
3.7.3.	Fooducate.....	22
3.7.4.	Sleep Cycle alarm clock .....	22
3.8.	Senioři.....	22
3.8.1.	BIG Launcher .....	23
3.8.2.	Med Helper Pill Reminder.....	23
3.8.3.	AroundME .....	23
3.9.	Shrnutí .....	23
4.	Cíle práce.....	25
4.1.	Průzkum využití aplikací .....	25
4.2.	Cíle práce .....	26
4.3.	Návrh aplikací.....	27
5.	Použité technologie .....	32
5.1.	Android.....	32
5.2.	QR kód.....	32

5.2.1.	Knihovna Zxing .....	33
5.3.	OpenCV knihovna .....	34
5.4.	ColorThief knihovna.....	35
5.5.	SQLite databáze.....	37
5.6.	Grafické uživatelské rozhraní .....	38
5.7.	Hlasový průvodce .....	40
5.8.	JavaFX .....	41
6.	Vývoj mobilní aplikace .....	43
6.1.	Struktura projektu v Android Studiu .....	43
6.2.	Struktura Android aplikace.....	45
6.2.1.	Obecné třídy.....	45
6.2.2.	Databázové třídy .....	49
6.2.3.	Pomocné třídy .....	50
7.	Vývoj desktopové aplikace .....	52
7.1.	Main.java .....	52
7.2.	Controller.java .....	52
8.	Testování .....	53
8.1.	Testování Android aplikace.....	53
8.2.	Testování JavaFX aplikace .....	54
8.3.	Uživatelské testy .....	54
8.3.1.	Modul pro rozeznání barvy.....	55
8.3.2.	Modul pro čtení QR štítků .....	56
8.3.3.	Modul s léky .....	57
9.	Rozšíření aplikace .....	59
10.	Závěr.....	60
	Literatura .....	
	Slovník zkratk.....	

Seznam příloh.....	
Příloha A: Rozhovor s třídní učitelkou.....	
Příloha B: Dopis organizacím .....	
Příloha C: Odpovědi na dopis .....	
Příloha D: Uživatelská dokumentace .....	
Příloha E: Diagram tříd .....	
Obsah CD .....	



# 1. Úvod

Mobilní zařízení mohou v dnešní době používat i lidé s nějakým zdravotním postižením. Vzhledem k rychlému vývoji technologií klesá i cena takových zařízení a stávají se tak více přístupná všem lidem. Avšak nejen finance jsou faktorem, díky kterým by byla daná technologie přístupnější.

Stačí, aby měl člověk oslaben pouze jeden ze svých smyslů, a již může dojít k zabránění použití určitého přístroje či poskytnutí služby. Většina výrobců však nezapomíná ani na tyto osoby s omezením a zabudovávají asistivní technologie<sup>1</sup> přímo do svých přístrojů, aby byly dostupné co nejvíce lidem.

Operační systémy velkých firem, jako je Apple a Microsoft, mají dnes zabudovaný software pro předčítání textu z obrazovky. Objevují se zdarma dostupné open-source text-to-speech<sup>2</sup> technologie financované společnostmi Google, Mozilla, aj. S těmito aspekty také souvisí neustálé zlevňování mobilních zařízení. Stejně tak je to i s aplikacemi, které jsou pro tato zařízení vyvíjeny. Dříve měly osoby se zdravotním postižením více malých přístrojů s embedded<sup>3</sup> systémy. S příchodem smartphonů je však možnost vyvinout aplikace s funkcemi těchto malých přístrojů a poté je nainstalovat do jednoho smartphonu. V současné době žije v České republice okolo jednoho milionu osob se zdravotním postižením [1]. Jelikož roste i počet osob vlastnících smartphone, je důležité věnovat pozornost vývoji aplikací pro takové osoby.

Zaměření diplomové práce se týká možností operačního systému Android z hlediska podpory osob s různými zdravotními postiženími. V analýze je nejprve popsána kategorizace postižení spolu se souvisejícími již existujícími aplikacemi pro pomoc postiženým. Dále byly stanoveny cíle práce, s čímž souviselo provedení průzkumu využití aplikací v podobě zaslání dotazníku na organizace sdružující osoby s různými zdravotními postiženími.

---

<sup>1</sup> Souhrnné označení pro pomůcky, které pomáhají zlepšit fyzické nebo duševní funkce

<sup>2</sup> TTS – programy produkující řeč z textu psaného běžným jazykem

<sup>3</sup> Vestavěný, zabudovaný systém

Byly vybrány vhodné funkcionality Android aplikace pro usnadnění některých běžných každodenních činností lidí se zrakovým postižením, tedy aplikace pro rozeznání předmětů pomocí jejich označení štítky a pro rozeznání dominantní barvy z obrázku. Tato aplikace byla poté navržena, realizována a byla ověřena její funkcionality. Dále byla vytvořena podpůrná aplikace pro systém Windows umožňující vytisknutí výše zmíněných štítků. Na závěr byla navržena další možná rozšíření.

## **2. Analýza**

V této kapitole je vysvětlen pojem asistivní technologie a jsou popsány druhy těchto technologií. Kapitola slouží pro seznámení a uvedení do problematiky různých postižení. Provedl jsem analýzu těchto postižení a jejich následnou kategorizaci. Každá tato kategorie obsahuje definici postižení a případně také konkrétní příklad postižení.

### **2.1. Asistivní technologie**

Asistivní technologie je souhrnné označení pro pomůcky, které pomáhají zlepšit fyzické nebo duševní funkce osobám, které mají tyto funkce z různých důvodů sníženy. Pod pojem asistivní technologie lze zahrnout nejen tyto pomůcky samy o sobě, ale i služby spojené s jejich poskytováním [2]. Osobám se zdravotním postižením mohou mírnit dané onemocnění, bránit zhoršení stavu, zlepšit schopnost člověka učit se, nebo dokonce nahradit chybějící končetinu.

Kromě pomůcek, jako je výše zmíněná náhrada končetiny, představují širokou oblast asistivních technologií pomůcky usnadňující komunikaci s počítačem. Vzhledem k tomu, že komunikace pomocí výpočetní techniky je dnes již běžná záležitost, je velmi podstatné věnovat této oblasti asistivních technologií značnou pozornost.

Asistivní technologie pokrývají širokou oblast, avšak dají se rozdělit do následujících kategorií (viz Tabulka 2.1). U každé kategorie je vypsán příklad pro usnadnění práce s počítačem.

Porucha	Příklad
Poruchy vidění	Software zvětšující část obrazovky
	Zvětšení textu na monitoru
	Softwarové čtečky
	Hlasový vstup textu, hlasový průvodce
Poruchy sluchu	Videokonferenční hovor
	Možnost zesílení zvukového výstupu
Poruchy pohybových schopností	Ergonomicky modifikované klávesnice, trackbally
	Přepínač ovládaný mrknutím víčka, pohybem končetiny
	Eyegaze <sup>4</sup>
	Komunikace pomocí snímání pohybů nohou, úst
Duševní poruchy	Čtečky dokumentů (neschopnost soustředit se)
	Hlasový průvodce
	Slovníky a kontrola pravopisu

**Tabulka 2.1: Kategorie asistivních technologií**

## 2.2. Kategorizace postižení

Člověk, jenž je postižen dlouhodobou (např. měsíce či roky) nemocí, je zařazen do tzv. zdravotní skupiny [3]. Následující kapitola tyto druhy postižení kategorizuje a každá tato kategorie je popsána.

<sup>4</sup> Klávesnice na obrazovce ovládaná pomocí oka, jehož pohyb je sledován pomocí kamery

### **2.2.1. Zrakové postižení**

Zrakově postižení lidé - nevidomí, jsou invalidní osoby mající různé druhy a stupně snížených zrakových schopností. Ať už se jedná o oční choroby, vady a jiná poškození s charakteristickými následky pro vidění, u těchto lidí je určitým způsobem omezena schopnost vnímat, rozlišovat nebo představovat si prostředí. Vidění rozlišujeme na centrální a periferní. Pomocí centrálního vidění jsme schopni vnímat barvy a detaily, periferní vidění nám pak umožňuje vnímat prostor a orientovat se v něm.

Takoví lidé jsou pak omezeni v činnostech v běžném životě. Do této skupiny nezahrnujeme např. osoby s dioptrickými brýlemi a pro zdůraznění takového rozdílu použijeme pojem těžce zrakově postižení. Tuto skupinu můžeme dále rozdělit do podskupin - na nevidomé a slabozraké.

Pro objektivnější diagnostiku musíme také zkoumat další zrakové funkce, jako např. kontrastní citlivost, schopnost rozlišovat barvy, vnímání hloubky, schopnost lokalizovat, sledovat předměty v pohybu apod.

### **2.2.2. Sluchové postižení**

Ať už se lidé s tímto postižením již narodili, nebo se sluchová vada projevila během života či ve stáří, v České republice je zhruba 0,5 milionů nedoslýchavých (mohou rozlišit hlásky) a neslyšících (nepoznají sluchem jednotlivé hlásky) lidí [4]. Sluchové postižení se dále specifikuje dle druhu, stupně závažnosti, doby vzniku hluchoty či osobnostních předpokladů.

Schopnost komunikace lidí s tímto postižením bývá v mluvené řeči obtížná - jednak se naskytá problém mluvenou řeč vnímat, jednak je často z důvodů nemožnosti sluchové kontroly pro tyto osoby nelehké mluvenou řeč produkovat. Postupným cvičením lze však tyto schopnosti vylepšit, úspěchy však nesouvisí s inteligencí, ale spíš s jistým nadáním.

Tito lidé komunikují většinou pomocí znakového jazyka, který v České republice používá 7 000 až 10 000 uživatelů. Komunikovat lze také pomocí odezírání ze rtů, ale to bývá nespolehlivé a je obtížné si tuto metodu osvojit. Jako kompenzační

pomůcka je pro tyto osoby často používáno naslouchátko zesilující zvuk vstupující do sluchovodu.

Český znakový jazyk je přirozený a plnohodnotný komunikační systém tvořený specifickými vizuálně-pohybovými prostředky, tj. tvary rukou, jejich postavením a pohyby, mimikou, pozicemi hlavy a horní částí trupu [5]. Často se vyskytuje také termín *znaková čeština*, což je umělý jazykový systém, který využívá gramatické prostředky českého jazyka. Pokud však neslyšící neovládá dobře gramatiku mluveného jazyka, těmto umělým systémům mnohdy nemusí rozumět.

### **2.2.3. Tělesné postižení**

Tělesné postižení je zdravotní postižení definované tělesnou odchylkou jedince omezující jeho pohybové schopnosti, což přímo ovlivňuje jeho kognitivní, emocionální a sociální výkony. „Jsou narušeny role, které postižený ve společnosti zastává: soběstačnost, schopnost cestovat, partnerská a rodinná role, pracovní a zájmová činnost.“[6]

Tato postižení mohou být buď vrozená vznikající během těhotenství či porodu (např. mozková obrna), či získaná (např. amputace končetiny po nehodě) a na rozdíl od vrozených mohou vzniknout v kterémkoliv období života.

### **2.2.4. Mentální postižení**

Mentálním postižením je nazýváno trvalé snížení rozumových schopností, které vznikne v důsledku organického poškození mozku. Částečný vliv mohou mít také specifické generické příčiny. Celkové procento osob s takovýmto postižením je odhadováno na 3% z celkové populace [7]. Jedná se o trvalý stav, který je klasifikován do šesti základních kategorií:

- Lehká mentální retardace (IQ 50 - 69)
- Středně těžká mentální retardace (IQ 35 - 49)
- Těžká mentální retardace (IQ 21 - 34)

- Hluboká mentální retardace (IQ nižší než 20)
- Jiná mentální retardace (nelze určit stupeň intelektové retardace pomocí obvyklých metod)
- Nespecifikovaná mentální retardace (mentální retardace prokázána, ale nelze zařadit do některé z kategorií)

U osob s takovýmto postižením (nejen u dětí) je v rámci výchovy a vzdělávání potřeba neustále opakovat a prohlubovat znalosti a dovednosti, což se stává celoživotním procesem. Základy jsou však v rodině a v přístupu rodičů k takovému dítěti, pro něž je obvykle obtížné vytrvat, jelikož výsledky se dostávají velmi zvolna a pokroky jsou takřka nepozorovatelné.

Pro přiblížení problematiky vzdělávání těchto dětí jsem využil možnosti navštívit speciální základní školu v mém rodném městě v Blatné a promluvil jsem si o tomto druhu výchovy s jednou ze třídních učitelek (viz Příloha A). V souvislosti s touto návštěvou níže uvádím § 33 ze Zákona o soustavě základních škol, středních škol a vyšších odborných škol, který se týká pomocných škol.

„Pomocná škola vychovává a vzdělává obtížně vzdělavatelné žáky s takovými nedostatky rozumového vývoje, pro které se nemohou vzdělávat ani ve zvláštní škole, jsou však schopni osvojit si alespoň některé prvky vzdělání. Obsah výchovně vzdělávací činnosti se zaměřuje na vypěstování návyků sebeobsluhy, osobní hygieny a na rozvíjení přiměřených poznatků a pracovních dovedností s předměty denní potřeby.

*(Zákon č. 29/1984 Sb. novely č.258/1996 Sb., § 33, odstavec 1)*

V základních školách speciálních se vzdělávají zpravidla žáci s mentálním postižením středního a těžšího stupně. Jejich vzdělávání vyžaduje odbornou speciálně pedagogickou péči a vhodně upravené podmínky – malý počet žáků ve třídě, třídy přizpůsobené jejich potřebám, speciální učebnice a pracovní sešity, přizpůsobený časový rozvrh a především klidné, nehlukné a nestresující prostředí, poskytující pocit bezpečí a jistoty a umožňující koncentraci žáků na školní práci.

Těžiště pedagogické práce spočívá ve výchovné a vzdělávací činnosti, zaměřené

na poskytování elementárních vědomostí, dovedností a návyků, potřebných k uplatnění v praktickém životě a na rozvoj duševních i tělesných schopností žáků, založené na důsledném respektování jejich individuálních zvláštností.

Současné pojetí vzdělávání je založeno na otevřenosti systémů komplexní péče o žáky se zdravotním postižením a znevýhodněním. Za žáky se zdravotním postižením se považují osoby s mentálním, tělesným, zrakovým nebo sluchovým postižením, s vadami řeči, se souběžným postižením více vadami, s autismem, s vývojovými poruchami učení nebo chování. Za žáky se zdravotním znevýhodněním se považují osoby zdravotně oslabené, dlouhodobě nemocné a osoby s lehčími zdravotními poruchami vedoucími k poruchám učení a chování.

Po zpracování výše zmíněného rozhovoru jsem přemýšlel, jak by bylo možné danou aplikaci realizovat. Určitě by bylo vhodné důkladně nastudovat zmíněnou metodu Sfumato tak, aby byla pro učitele aplikace přínosná, což by muselo být také podloženo delší spoluprací a testováním na žácích. Při učení technikou Sfumato "splývavé čtení" se zabrání dvojitému čtení, dítě čte plynule a s porozuměním, po stránce fyziologické se tvoří "půda" pro správné dýchání a umístění tónu v dutině ústní [8].

Jelikož však v rodném městě nepobývám často, abych byl poblíž této školy a také z důvodu, že z následné další vzájemné komunikace jsem se dozvěděl, že škola nemá peníze na mobilní zařízení s operačním systémem Android (ať už smartphone či tablet), od takovéto aplikace jsem upustil.

### **2.2.5. Vnitřní postižení**

Vnitřní postižení nemusí být na první pohled vidět, přesto může být velice vážné. Takto handicapovaných je z celkového počtu různě postižených osob zhruba 35% [9]. Mezi vnitřní postižení řadíme nemoci oběhové soustavy, endokrinní nemoci (např. diabetes), epilepsie a alergie [10].



## 2.2.6. Duševní nemoci

Takováto onemocnění mohou mít různé příčiny. Na duševní zdraví má vliv genetika, výchova, okolní prostředí a různé životní události. Vztah těchto aspektů k míře ovlivnění onemocnění je však stále předmětem zkoumání a bývá často diskutován. Léčba těchto poruch probíhá buď pomocí psychoterapií, léků, úpravou životního stylu, atp.

Název oddílu	Příklad
Organické duševní poruchy	Psychóza, demence
Poruchy způsobené užíváním psychoaktivních látek	Syndrom závislosti, akutní intoxikace
Schizofrenie, schizotypální poruchy	Paranoidní schizofrenie
Afektivní poruchy (poruchy nálady)	Deprese
Neurotické a stresové poruchy	Sociální fobie, panická porucha
Syndromy poruch chování, spojené s fyziologickými poruchami a somatickými faktory	Mentální anorexie, nespavost
Poruchy osobnosti a chování u dospělých	Pyromanie, paranoidní porucha osobnosti
Mentální retardace	Hluboká mentální retardace
Poruchy psychického vývoje	Specifické vývojové poruchy řeči a jazyka, poruchy čtení, atypický autismus
Poruchy chování a emocí se začátkem obvykle v dětství a v dospívání	Porucha aktivity a pozornosti, tiky, kocktavost
Neurčená duševní porucha	Duševní porucha, jinak neurčená

**Tabulka 2.2: Duševní poruchy**

Dle publikace Mezinárodní klasifikace nemocí, která je v České republice kodifikací systému označování a klasifikace lidských onemocnění, poruch a dalších příznaků [11], jsou duševní poruchy rozděleny do jedenácti oddílů (viz Tabulka 2.2).

### **2.2.7. Kombinovaná postižení**

Tato skupina zahrnuje osoby s jedním a více postižení z výše zmíněných, například hluchoslepota, mentální postižení se sluchovou vadou, či jiné kombinace vad tělesných a smyslových. Zvláště u dětí s mentálními poruchami je důležité rozvíjet jejich motorické schopnosti, aby byly tyto děti později schopny zvládnout základní osobní potřeby.

### **2.2.8. Senioři**

Ačkoliv se v tomto případě nejedná o druh postižení, senioři bývají často ve společnosti znevýhodněni svými fyzickými či psychickými schopnostmi, které se postupem věku u většiny takových lidí zhoršují. S postupem stáří dochází ke zhoršení pohybových schopností a paměti. Se zhoršením paměti může souviset také náhlá dezorientace. Další nebezpečnou situací může být náhlá nevolnost a v takovýchto případech je důležitá rychlá první pomoc. S narůstajícím věkem může také dojít ke zhoršení zraku, což vede k problémům se čtením textu z obrazovky mobilního zařízení.

## 3. Možnosti Androidu

V této kapitole je uvedeno, jaké možnosti nabízí operační systém Android jako takový v souvislosti s podporou osob se zdravotními postiženími. Provedl jsem analýzu užitečných aplikací zaměřených na podporu pomoci těchto osob. Tyto aplikace jsem vyhledával online, pozornost jsem věnoval článkům vydaných na webových stránkách sdružující handicapované osoby. Pokud to bylo možné, danou aplikaci jsem nainstaloval na svém mobilním zařízení a otestoval její použití.

### 3.1. Podpora v Androidu

Společnost Google se, stejně jako většina velkých IT společností, věnuje otázce podpory dostupnosti technologií osobám se zdravotními postiženími natolik, že má pro tuto oblast speciální tým lidí nazvaný *Accesible Team*. Tento tým se zabývá problémem, jak zpřístupnit dostupné informace uživatelům s různými postiženími, zejména však se zrakovým postižením, barvoslepostí a sluchovým postižením. Tyto problémy pak řeší vývojem pomocných API a webových služeb pro platformy Google Android a Chrome OS. Udržují kontakt se skupinami sdružující osoby s takovými postiženími, zpracovávají jejich připomínky, které poté využívají pro zlepšování těchto produktů.

Jedním nástrojem, jak společnost pomáhá vývojářům vylepšit přístupnost jejich aplikací, je *Accessibility Scanner*. Tento nástroj slouží k tomu, aby určil, jak je aplikace přístupná, přehledná a v případě možností na vylepšení ukáže vývojáři, jak ji lze vylepšit (například zvětšit kontrast u určitého nadpisu).

S příchodem verze Android N je dostupné vylepšené nastavení zobrazení, kdy uživatel může zapnout možnost přiblížení části obrazovky po třech kliknutích na danou oblast, měnit velikost písma a obecně všech objektů na obrazovce a také je zde dostupná čtečka obrazovky *TalkBack*. Tato aplikace byla dříve dostupná jako doplňková k doinstalování a od této verze je v zařízeních integrována. Poskytuje hlasovou zpětnou vazbu a uživatel je tak schopen používat zařízení i bez dívání se na obrazovku.

Společnost Google také dále vyvíjí aplikaci *Voice Access*, která je nyní ve fázi testování. Tato aplikace umožňuje osobám, které mají z nějakého důvodu problémy

používat dotykovou obrazovku, ovládat zařízení pomocí hlasových příkazů. Pokud je tato aplikace aktivní, umožňuje například ovládat obrazovku s více objekty tak, že ke každému takovému objektu umístí číslo a tím jej může uživatel identifikovat hlasem. Zároveň se v oznamovací liště zobrazují příkazy, které uživatel do zařízení zadává. Tato aplikace je vhodná jak pro osoby se zdravotními postiženími, tak jako náhrada za hands-free zařízení v automobilech.

V souvislosti s předchozí kapitolou následuje přehled aplikací, které nějakým způsobem ulehčují uživatelům pracovat s mobilním zařízením s operačním systémem Android. V případě výskytu aplikace s vysokou rozšířeností či vysokým potenciálem jsou zde zahrnuty i některé aplikace pro operační systém společnosti Apple iOS.

## **3.2. Zrakové postižení**

Od verze Android Lollipop (přesněji od verze 5.0) lze v systému Android nastavit různé módy displeje umožňující barvoslepým lidem přizpůsobit obrazovku dle své diagnostiky barvosleposti.

Color Space Correction obsahuje 6 typů korekce barvosleposti:

- Deuteranomaly (červeno-zelená)
- Protanomaly (červeno-zelená)
- Tritanomaly (modro-žlutá)
- Deuteranopia (zelená)
- Protanopia (červená)
- Tritanopia (modrá)

Pokud uživatel zvolí některý z těchto druhů barvosleposti, změní se podle zvoleného typu uživatelské rozhraní mobilního zařízení. Tato změna se projeví také na veškerém zobrazovaném obsahu, jako jsou fotografie, videa, aplikace a hry.

Existuje několik druhů mobilních aplikací, které se snaží takovými lidem pomoci lépe komunikovat, cestovat, či trénovat zrak a zlepšit si tak schopnost fungovat v běžném životě.

### **3.2.1. Braille Helper**

Aplikace pro operační systém Android, která má několik módů. Lze si prohlédnout a přehrát abecedu spolu s příslušným znakem v Brailleově písmu, dále je zde mód procvičování, kdy uživatel zadává na klávesnici slova. V interaktivním módu lze pomocí výběru teček zjišťovat původ písmene a nechybí zde také překladač slov do Brailleova písma. Aplikace je zdarma.

### **3.2.2. ColorBlindness SimulateCorrect**

Tato aplikace simuluje a upravuje barvoslepost v reálném čase za použití integrované kamery v mobilním zařízení. Jsou zde zastoupeny tři druhy barvosleposti - tritanopia, protanopia a deuteranopia. Dále jsou zde dva módy, a to normální, kdy je simulace po celé obrazovce displeje a porovnávací, kdy je obraz rozpuřen a v jedné polovině vidíme normální obraz a v druhé polovině obraz simulovaný. Aplikace je pro zařízení s operačním systémem Android a je zdarma.

### **3.2.3. Be My Eyes**

Jedná se o aplikaci pro operační systém iOS spojující slepé osoby s dobrovolnými pomocníky. Pokud slepý člověk potřebuje v daný moment vysvětlit či popsat situaci, ve které se právě nachází, vyfotí ji, dobrovolník se na obrázek podívá a danou situaci vysvětlí hlasovou zprávou.

Aplikace je zdarma a díky široké komunitě dobrovolníků dle webových stránek vydavatele tímto způsobem dosud pomohla slepým lidem ve více jak 100 000 případech [12].

### **3.2.4. Learn Braille Alphabet**

Aplikace pro operační systém iOS, která má pomoci procvičovat Brailleovu abecedu. Je však nepoužitelná pro slepé osoby, jelikož ty při rozeznání takových písmen používají hmat, což není na smartphonu možné. Její cena jsou \$2.

### 3.3. Sluchové postižení

Co se týče sluchových postižení, společnost Google vyvíjí pro platformu Android *Text-to-Speech* engine. Jedná se o engine umožňující aplikacím v mobilním zařízení s operačním systémem Android převádět text na řeč. Pro tyto účely je využita třída *TextToSpeech*, která je k dispozici od Android verze API 4 a výše. Instance této třídy může být použita k přehrání až poté, co je dokončena její inicializace. Pokud bychom chtěli být o tomto stavu informováni, je zapotřebí implementovat listener *TextToSpeech.OnInitListener*. Aplikace je aktualizována a spolu s novými verzemi je přidávána podpora pro další jazyky.

Dostupné jazyky: angličtina (Austrálie), angličtina (Indie), angličtina (USA), angličtina (Velká Británie), bengálština (Bangladéš), dánština, finština, francouzština, hindština, indonéština, italština, japonština, kantonština (Hongkong), korejština, maďarština, mandarínština (Čína), mandarínština (Tchaj-wan), němčina, nizozemština, norština, polština, portugalská (Brazílie), ruština, španělština (Španělsko), španělština (USA), thajština a turečtina.

Engine sice umožňuje přehrát i nepodporovaný jazyk, výsledná kvalita je však nedostačující a na pokraji srozumitelnosti z důvodu synteticky znějícího hlasu. Čeština zatím patří mezi nepodporované jazyky. V únoru roku 2016 jsem poslal dotaz prostřednictvím e-mailu na stránky podpory od vývojářů Google Text-to-Speech, ve kterém jsem se chtěl dozvědět, zda má společnost v plánu přidat do dostupných jazyků i český jazyk, a pokud ano, zda-li bych mohl dostat bližší informace o datu tohoto přidání. Na tento e-mail mi však nikdo dosud neodpověděl (datováno k dubnu roku 2016).

#### 3.3.1. RogerVoice

Aplikace vyvinutá pro operační systémy iOS a pro operační systém Android, pro který je dostupná od verze Android 4.1. RogerVoice převádí hovor do textu. Neslyšící pak může volat tak, že sleduje displej a na něm se překládá hovor od druhého účastníka komunikace. Využívá také TTS, čímž se stává také vhodnou aplikací pro osoby, které nemohou mluvit a odpovědět do mikrofону. Odpověď napíše na klávesnici a ta se v přednastaveném jazyce přehraje.

RogerVoice využívá internetové spojení Wi-Fi, je však možné využít i mobilní síť (3G, 4G). Aplikaci lze vyzkoušet v trial verzi zdarma, pro častější použití lze pak aplikaci pořídit v různých variantách, například za cenu 1,99€ na měsíc, jenž vystačí na 40 minut hovoru.

### **3.3.2. The Deaf And Hearing Impaired**

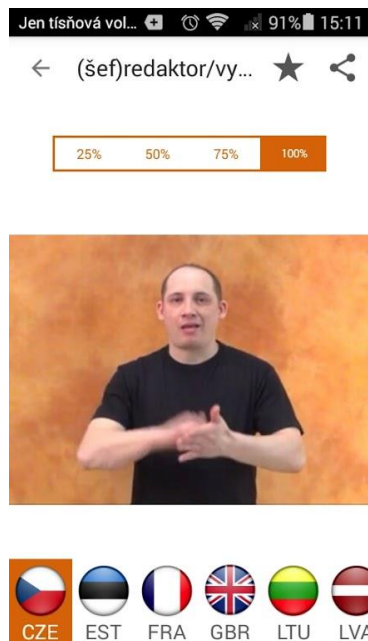
Aplikace pro operační systém Android, jejíž hlavní funkcí je převod hlasu na text a opačně. Nemá však větší možnosti pro nastavení hlasu a jazyka, během testování se některé nadpisy zobrazovaly nečitelně a i když tato aplikace používá pro rozeznání hlasu rozhraní společnosti Google, její výsledky byly kvůli těmto poznatkům nepříznivé. Aplikace je zdarma.

### **3.3.3. Deaf - Hearing chat device**

Android aplikace, jenž byla vyvinuta jako chatovací aplikace mezi slyšící osobou, která komunikuje přes mikrofon a poté je hlas převeden na text a zobrazen osobě neslyšící. Tato osoba poté komunikuje pomocí klávesnice. Aplikace funguje mezi dvěma mobilními zařízeními přes rozhraní Bluetooth a je zdarma.

### **3.3.4. Spread Signs**

Jedná se o slovník znakové řeči s více jak 200 000 znaky v různých jazycích. Uživatel při prvním spuštění vybere jeho rodný jazyk, ve kterém bude zadávat slova a po vyhledání potřebného slova se mu zobrazí v daném jazyce krátké video osoby, která daný znak předvede ve znakové řeči. Dále může uživatel tento jazyk změnit a podívat se, jak toho slovo vypadá v jiném jazyce.



**Obrázek 3.1: Aplikace Spread Signs**

Spread Signs jsem testoval na svém zařízení, její výhodou je rozmanitost jazyků, včetně českého. Pomocí vyhledávače lze zadat libovolné slovo, kdy je po výběru shodného slova zobrazeno video s osobou, která toto slovo ukáže ve znakové řeči (viz Obrázek 3.1). Aplikace je zdarma, avšak její funkce jako Prstová abeceda, Oblíbené a Dětské znaky je potřeba dokoupit. Ke správnému fungování je potřeba Wi-Fi síť, nebo datové připojení.

### 3.4. Tělesné postižení

Obecně je jako u ostatních zmíněných postižení snaha takové osoby začlenit do společnosti a ulehčit jim život, práci s mobilními zařízeními nevyjímaje. Společnost Google (stejně jako konkurenční společnosti) vyvíjí inteligentní osobní asistenty, kteří jsou dostupní z mobilních zařízení. V případě společnosti Google se jedná o službu Google Now, u Apple je to služba Siri, u Microsoft se jedná o službu Microsoft Cortana.

Služba Google Now byla poprvé zahrnuta do Androidu verze 4.1, která byla vydána 9. července 2012. V mobilním zařízení je dostupná v mobilní aplikaci Google (vyhledávač). Používá uživatelské rozhraní přirozeného jazyka (NLUI). To znamená, že



odpovídá na dotazy uživatele pomocí přeložení dotazu a vykonání požadavků na sadu webových služeb. Níže jsou uvedeny 2 aplikace, které se touto problematikou zabývají.

### **3.4.1. Commandr for Google Now**

Aplikace rozšiřuje použitelnost služby Google Now o několik příkazů, které dokáží přímo interagovat s mobilním zařízením, například zapnout svítilnu, Wi-Fi síť, nebo přečíst poslední SMS zprávu. Veškeré vestavěné příkazy lze libovolně nakonfigurovat, či přidat vlastní - to je však možné pouze po doinstalování aplikace Tasker.

Tato aplikace je zdarma, některé její funkce však vyžadují root<sup>5</sup> mobilního zařízení.

### **3.4.2. Tecla Access**

Tecla Access je souhrn nástrojů, které jsou speciálně navrženy pro osoby s tělesným postižením a mají tak problém ovládat svůj smartphone či tablet. Aplikace obsahuje klávesnici s podporou hlasu (skrz výše zmíněný Google Now) a umožňuje vzdáleně ovládat mobilní zařízení pomocí elektrického vozíku pro invalidy.

Aplikace je zdarma, avšak pro její použití je nutné si od společnosti Tecla pořídit také hardwarové zařízení Tecla Shield DOS za cenu \$349 a potřebné spínače, či joysticky, pohybující se okolo ceny \$100 za kus.

## **3.5. Mentální postižení**

Android systém dosud neobsahuje zabudovaný systém či aplikace pro umožnění práce se zařízením mentálně postiženým lidem.

---

<sup>5</sup> Proces vedoucí k umožnění ovládat zařízení s administrátorskými právy

### **3.5.1. Avaz - AAC App for Autism**

Jedná se o AAC<sup>6</sup> aplikaci vyvinutou pro děti mající problémy s mluvením, či komunikující pouze neverbálně. Aplikace je zaměřena pro výuku autistů, dětí s Downovým syndromem apod. Používá obrázky spolu se syntetizovanou řečí, pomocí čehož může postižené dítě vytvořit určitou zprávu. Obrázky a jejich popisy lze editovat, je zde také možnost vytvořit si vlastní, a to s pomocí databáze obrázků či pomocí vlastních fotografií.

Aplikace je vytvořena pro tablety a pro její instalaci je zapotřebí minimálně 2 GB volného místa. Její cena je 4 456,- Kč, lze nainstalovat i verzi Lite, která je však časově omezena - nikde jsem však nenalezl na jak dlouho.

### **3.5.2. Talkitt**

Pomocí mikrofonu zpracuje a přetlumočí řeč lidí s řečovými vadami do srozumitelné řeči. Nejedná se o klasickou aplikaci pro rozpoznání řeči, jsou zde dvě fáze, které proběhnou před samotným tlumočením. V první fázi se aplikace kalibruje - učí se, jak daný uživatel mluví, jaké jsou jeho vady a vytvoří se tak uživatelský slovník. V další fázi již dochází k samotnému rozpoznávání a následnému překládání do srozumitelné řeči.

Aplikace se objevila na trhu jako start-up projekt a byla velmi úspěšná na crowdfundingovém serveru Indiegogo, kde tento projekt splnil 200% požadované částky \$40 000 USD, celkově bylo vybráno \$87 035 USD. Aplikace však zatím nebyla uveřejněna (datováno k dubnu roku 2016).

## **3.6. Vnitřní postižení**

V této kapitole jsou zmíněny aplikace třetích stran věnující se tomuto postižení. Jde o postižení, která nemusí být na první pohled zřejmá, příkladem zde mohou být alergie. Více viz kapitola 2.2.5.

---

<sup>6</sup> Augmentativní a alternativní komunikace

### **3.6.1. MySugr Diabetes Logbook**

Jedná se o deník pro diabetiky, který umožňuje zaznamenávat hladinu cukru, sacharidy, zřehledňuje dávky inzulínu uživatele, eviduje léky a z těchto dat poté zobrazuje jednoduché analýzy. Aplikace umožňuje personalizovat hlavní obrazovku a zálohovat data. Placená verze má navíc schopnost extrahovat reporty ve formátech .pdf a .xls, které pak mohou využít lékaři těchto pacientů například pro změnu dávkování léků. Tuto aplikaci jsem z důvodů kompatibility nebyl schopen na svém mobilním zařízení otestovat, jelikož je její instalace pro Českou republiku omezena. Aplikace je zdarma.

### **3.6.2. My Epilepsy Diary**

Obdobná aplikace, která je však určena pro epileptiky. Záznam je synchronizován na server a uložen k příslušnému zaregistrovanému uživateli. Umožňuje také nastavit upomínky na užívané léky a ukládat fotografie k denním záznamům, což může být využito například pro pozdější zaznamenání spouštěče epileptického záchvatu, zachycení stavu léků, apod. Aplikace je zdarma. V březnu roku 2016 byla vydána nová verze a tato aplikace byla přejmenována na EpiDiary.

### **3.6.3. Find Me Gluten Free**

Aplikace pro vyhledávání restaurací, které podávají bezlepková jídla. Vyhledávání probíhá pomocí GPS modulu, či pouze dle zadané adresy. Zobrazuje také hodnocení daných restaurací a po zaregistrování lze zobrazit detaily vybrané restaurace. Při testování aplikace vyhledala šest takových restaurací v mém okolí (v Plzni). Aplikace je zdarma.

### **3.6.4. ZYRTEC® AllergyCast**

Tato aplikace je vhodná pro alergiky - monitoruje aktuální a předpovídá denní koncentraci pylu (tzv. pollen index). Pomáhá zaznamenávat alergické symptomy

uživatelé. Umožňuje nastavit upozornění, pokud se uživatel dostane do oblasti se zvýšenou koncentrací pylu, nepodařilo se mi však zjistit, na jakém základě jsou tyto oblasti určeny a kolik měřících stanic je zapojeno. Jelikož je aplikaci možno instalovat jen v některých zemích, mezi kterými však není Česká republika, nebyl jsem schopen ji nainstalovat na svém mobilním zařízení a otestovat. Aplikace je zdarma.

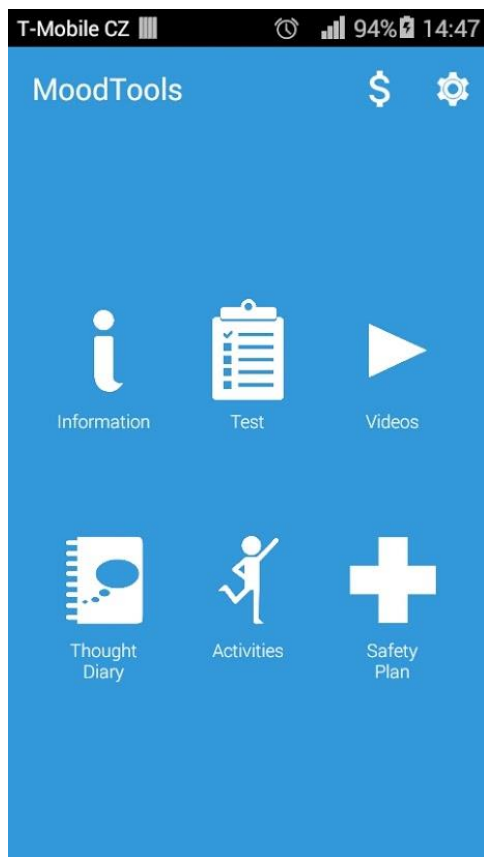
### **3.7. Duševní nemoci**

Android systém dosud neobsahuje zabudované aplikace pomáhající lidem s duševními nemocemi.

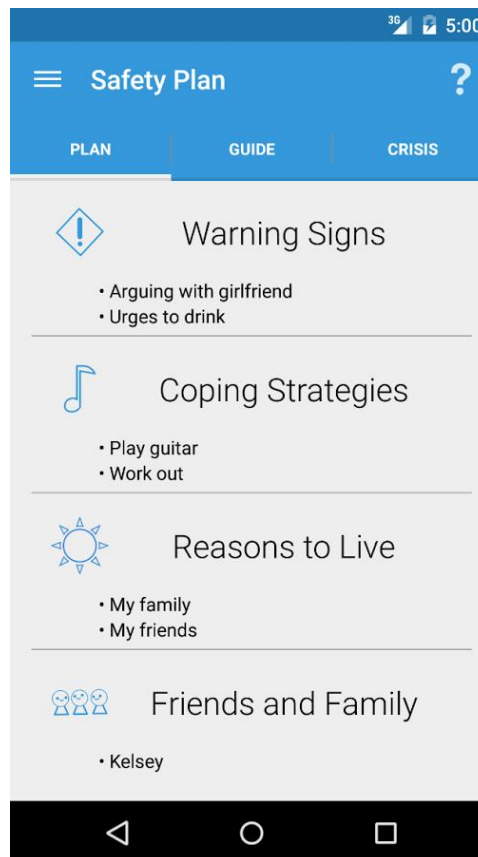
#### **3.7.1. MoodTools - Depression Aid**

MoodTools aplikace byla vyvinuta ve spolupráci s psychiatry a je určena pro pomoc lidem trpícím depresemi. Obsahuje několik modulů - analýza myšlenek, monitor aktivit, bezpečnostní plán, informace a pokyny pro svépomoc, video modul (motivační videa, aj.), test (dotazník pro přehled příznaků).

Aplikaci jsem testoval na svém zařízení. Vzhledem k tomu, že je v angličtině a obsahuje velké množství textů a informací, může tento fakt působit potíže při používání v českém prostředí. Úvodní stránka obsahuje rozdělení na moduly (viz také Obrázek 3.2):



**Obrázek 3.2: Menu aplikace MoodTools**



**Obrázek 3.3: Plán záchrany aplikace MoodTools**

1. Informace – obsahuje informace o tom, co je to deprese, jaké jsou její symptomy, typy, příčiny a způsoby léčení.
2. Test – dotazník, pomocí jehož odpovědí je určeno skóre, které představuje aktuální rozpoložení a v případě vysokého skóre je uživateli doporučena návštěva psychiatra.
3. Videá – zde jsou odkazy na meditační videa, videa se smyčkami uklidňujících zvuků a dále videa z přednášek konferencí TED Talks.
4. Diář myšlenek – zde má uživatel možnost zaznamenávat své myšlenky, jejich povahu a možná řešení.
5. Aktivity – tento modul umožňuje uložit proběhlé aktivity a zaznamenat pocity před vykonáním a po vykonání těchto aktivit.
6. Bezpečnostní plán (viz Obrázek 3.3) – zde je možno uložit své vlastní pocity, díky nimž by měl uživatel být schopen upustit od nějakého ubližování sama sobě (např. důvody žít, oblíbená místa, varovné signály, kontakty blízkých).

Dle portálu pro stahování aplikací Google Play má aplikace 100 000 – 500 000

stažení, čili je žádaná a věřím, že může být nápomocná lidem v tíživých situacích. Její nedostatek je však již zmíněná absence češtiny. Aplikace je zdarma.

### **3.7.2. Code Blue**

Aplikace, která má být nápomocná pro teenagery podléhající depresím, či děti potýkající se s šikanou. V případě nouze taková osoba stiskne tlačítko, kterým se vybraným osobám zašle aktuální poloha mobilního zařízení a pomoc tak může dorazit k dané osobě včas. Aplikace je v beta-verzi, její vydání zatím není určeno (datováno k únoru 2016).

### **3.7.3. Fooducate**

Jedná se o aplikaci, která má napomáhat ke zhubnutí. Shromažďuje informace o nutričních hodnotách produktů, nabízí různé varianty diet, umožňuje částečnou personalizaci aplikace, v placené verzi pak dokáže nabídnout sestavený jídelníček.

### **3.7.4. Sleep Cycle alarm clock**

Aplikace monitoruje pomocí akcelerometru fáze spánku uživatele a dle vyhodnocení nejvhodnější chvíle určí čas probuzení s ohledem na zadaný nejpozdější čas probuzení. Umožňuje zobrazení statistik, grafů fází spánku a přizpůsobit si způsob probuzení. Takový budík pak může mít vliv na psychické rozpoložení v aktuálním dni, jelikož napomáhá k lehčímu probuzení a pocitu odpočínutí. Aplikace je zdarma.

## **3.8. Senioři**

Operační systém Android umožňuje použití zjednodušených prostředí, která změni vzhled celého uživatelského prostředí mobilního zařízení. Kromě nich existuje také spousta aplikací, které jsou pro seniory vhodné. Některé z nich jsou uvedeny níže.

### **3.8.1. BIG Launcher**

Aplikace umožňuje používat smartphone seniorům a lidem s očními vadami a motorickými problémy. Jedná se o nahrazení uživatelského prostředí mobilního zařízení s Androidem verze 2.1 a vyšším. Jeho hlavními znaky jsou velké ikony, velké textové popisky, velký editor SMS a SOS tlačítko s geolokační funkcí. Aplikace stojí 250,- Kč včetně DPH.

### **3.8.2. Med Helper Pill Reminder**

Jedná se o aplikaci, která napomáhá k zřehlednění návštěv lékařů, užívání léků a jejich připomínání. Dále je zde také možnost vložit kontakty na jednotlivé lékaře, informace o užívaných lécích a lze také například zaznamenávat denně informace jako tep, tlak a teplotu. Umožňuje použití více profilů uživatelů a je zdarma.

### **3.8.3. AroundME**

AroundME identifikuje pomocí geolokačního zařízení polohu uživatele a umožní mu zobrazit nejbližší místa zájmu, např. nemocnice, čerpací stanice, restaurace, atd. Aplikace je zdarma.

## **3.9. Shrnutí**

Pro shrnutí kategorií postižení a zmíněných aplikací z předchozích kapitol je v Tabulce 3.1 uveden přehled jednotlivých postižení a k danému postižení je pak zmíněno několik zastupujících aplikací.

Typ postižení	Android aplikace	Komentář
Zrakové postižení	Braille Helper	Procvičení Braillova písma
	ColorBlindness SimulateCorrect	Simulace barvosleposti
	Be my eyes	Podpora dobrovolníků pro slepé
Sluchové postižení	Roger Voice	Neslyšící mohou volat, s využitím TTS
	Spread Signs	Slovník znakové řeči
Tělesné postižení	Tecla Access	Souhrn nástrojů pro osoby s tělesným postižením
Mentální postižení	Avaz	Výuka dětí s mentálním postižením
	Talkitt	Překladač hlasu lidí s vadou řeči
Vnitřní postižení	mySugr Diabetes Logbook	Deník pro diabetiky
	My Epilepsy Diary	Deník pro epileptiky
Duševní nemoci	MoodTools	Pomoc při depresi
	Sleep Cycle alarm clock	Inteligentní budík
Senioři	Med Helper Pill Reminder	Přehled o doktorech, lécích

**Tabulka 3.1: Přehled postižení a aplikací**



## 4. Cíle práce

Pro zvýšení reálného využití vytvářené aplikace jsem provedl nejprve průzkum využití aplikací ve formě rozeslání dotazníku. V této kapitole jsou dále stanoveny cíle práce a návrh aplikací.

### 4.1. Průzkum využití aplikací

Vytvořil jsem dotazník (viz příložené CD) a tento dotazník jsem rozeslal prostřednictvím e-mailu v září roku 2015 na organizace, které se nějakým způsobem podílely na podporování osob se zdravotním postižením, nebo které byly zainteresovány do výchovy dětí s handicapem. Celkem bylo ve zmíněném období osloveno 17 organizací pomocí 25 různých e-mailových adres textem, viz Příloha B. Avšak z tohoto počtu oslovených respondentů jsem se setkal s odpověďmi pouze od čtyř, viz Příloha C.

I přes malé množství odpovědí jsem se snažil odvodit nějaký závěr, který by mohl mít vliv na navržené funkčnosti. Z dotazníku, ale i během analýzy aplikací, jsem narazil na problém s absencí češtiny. Nemyslím si, že by na trhu chybělo velké množství aplikací s českou lokalizací, avšak pokud jde o takto specifické aplikace, jazyková bariéra je zde znatelná. Jako jeden z požadavků na funkčnost jsem tedy určil, že aplikace bude dostupná v českém jazyce.

Dále jsem se chtěl vyvarovat tomu, že bych kopíroval nápad nějaké jiné aplikace, a to mimo jiné z toho důvodu, že jsem nechtěl řešit problémy s vlastnictvím práv. Co se týče zaměření aplikace na určitou kategorii osob se zdravotním postižením, měl jsem jasno poté, co jsem navštívil exhibici s názvem „Neviditelná výstava“ v Praze.

Tato ojedinělá výstava se snaží přiblížit svět nevidomých tak, že návštěvník je prováděn několika místnostmi v absolutní tmě a během této cesty se setká s několika situacemi, které nás jinak obklopují celý život a handicap nevidomých si ani pořádně neuvědomujeme. Jsou zde místnosti jako obývací pokoj, místnost se sochami, je zde také místnost se simulací rušné ulice, lesa a ke konci je bar, kde je možnost objednat si pití a vyzkoušet si jej poslepu zaplatit. Po celou dobu návštěvy je k dispozici slepý průvodce, se kterým jsem si na konci výstavy popovídal ohledně jeho zkušenosti

s mobilními zařízeními. Dle jeho slov není těžké aplikace ovládat poslepu, pokud je člověk zná, či má možnost je ovládat pomocí hlasového průvodce. Takových aplikací je však málo a na nevidomé se v tomto ohledu zapomíná.

Dále jsme se snažili společně vymyslet nějaké funkčnosti, které by využil v reálném životě. Z jeho záliby ve vaření vyvstal nápad na aplikaci, která by snímala štítek z kořenek a přečetla název této kořenky nahlas. Ne vždy je totiž možné mezi sebou jednotlivá koření rozeznat podle čichu. Dále se zmínil o problému při praní prádla, kdy není bez cizí pomoci schopen prádlo roztřídit dle barev a aplikace na rozpoznání barev by tento problém vyřešila.

Rozhodl jsem se, že tyto dva nápady spojím v jednu aplikaci a budou odděleny na jednotlivé moduly. Po konzultaci s vedoucím diplomové práce byl přidán ještě třetí modul a to modul kombinující načítání štítků, přesněji štítků na krabičkách od léků, kdy bude možnost nastavit frekvenci užívání léku, notifikace pro užití a možnost označení léku jako „užitý“.

## **4.2. Cíle práce**

Cílem práce je vytvořit aplikaci vhodnou pro použití nevidomými lidmi (případně lidmi se zrakovými potížemi), která bude sestávat ze tří modulů. První modul bude pomocí fotoaparátu v mobilním zařízení snímat štítek na předmětu, přiřazovat k němu hlasové poznámky a ty poté číst uživateli. Druhý modul bude zaměřen na užívání léků a uživateli bude umožněno určit interval užívání a zjistit, zda si daný den již lék vzal. Třetí modul bude pomocí fotoaparátu v mobilním zařízení rozeznávat základní barvy. Bude potřeba také vytvořit aplikaci, která umožní vytvoření a vytištění těchto štítků. Tato aplikace bude spustitelná na operačním systému Windows.

Aby aplikace umožňovala ovládání nevidomými osobami, bude dbán důraz na jednoduché uživatelské rozhraní a bude obsahovat hlasového průvodce. Aplikace bude v češtině.

### 4.3. Návrh aplikací

V rámci diplomové práce bude vytvořena mobilní aplikace pro operační systém Android a desktopová aplikace pro operační systém Windows.

Aplikace na mobilní zařízení bude využívat lokální databázi SQLite (více v kapitole 5.5), kde budou uložena data týkající se uložených štítků. Tyto štítky budou realizovány pomocí QR kódu (více v kapitole 5.2). Bude určen formát, ve kterém budou hodnoty těchto štítků zpracovávány. Audio poznámka daného štítku bude uložena v externím úložišti mobilního zařízení.

Aplikace bude rozdělena na tři moduly, toto rozdělení bude zřetelné i co se týče uživatelského rozhraní. Dále zde bude možnost rozeznat dominantní barvu z fotografie pořízené pomocí fotoaparátu mobilního zařízení. Tyto funkcionality budou provázeny nahranými hlasovými poznámkami, díky nimž se budou moci v aplikaci lépe orientovat uživatelé s případnými zrakovými potížemi. Výsledným aplikacím jsem dal jednotný název *MyPointer*.

Čtečka štítků	<ul style="list-style-type: none"><li>- Načtení štítku</li><li>- Uložení nové poznámky ke štítku</li><li>- Nahradit poznámku novou</li><li>- Smazání štítku</li></ul>
Připomínání léků	<ul style="list-style-type: none"><li>- Načtení léku (štítku)</li><li>- Uložení nové poznámky k léku</li><li>- Určení intervalu užití léku</li><li>- Označení užití léku</li><li>- Nahradit poznámku k léku novou</li><li>- Smazání léku</li></ul>
Rozpoznávání barev	<ul style="list-style-type: none"><li>- Rozpoznání barvy</li><li>- Přečtení rozpoznané barvy</li></ul>
Windows aplikace	<ul style="list-style-type: none"><li>- Vytvoření a vytisknutí štítku</li></ul>

**Tabulka 4.1: Navrhované funkce vytvářené aplikace**

Uložení štítků zmíněné v Tabulce 4.1 bude probíhat pouze lokálně a to tak, že daná audio poznámka bude uložena v externím úložišti mobilního zařízení. To z toho

důvodu, pokud by ke stejnému štítku měli dva uživatelé nahranou jinou audio poznámku, mohlo by mezi nimi dojít ke zmatení. Obdobně tak bude i u samotné textové poznámky, která bude uložena lokálně v databázi.

Kvůli zjednodušenému ovládní nevidomou osobou bude možné nastavit zmíněný interval užití léku pouze v hodinách. Uživateli však bude umožněno nastavit rozmezí až 72 hodin. Aby však uživatel nemusel tuto maximální hodnotu nastavovat po jedné, bude tento výčet zvolen s následujícím rozložením hodin: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16, 20, 24, 32, 48, 72. Na základě této frekvence bude také uživatel informován pomocí notifikací, že si má daný lék vzít.

Co se týče přepočítávání doby k užití léku, měl jsem dvě možnosti, jak k tomuto problému přistoupit. Bylo možné dobu dalšího užití přepočítat ihned po jejím vypršení, avšak zvolil jsem druhou možnost - vypočtení doby dalšího užití léku až poté, co si uživatel vezme daný lék. Důvodem bylo, že při první možnosti by mohla nastat situace, kdy by uživatel na lék zapomněl a tudíž si musel vzít daný prášek například dvakrát za den. Takové situace se však liší dle druhu léku a pokud by uživatel opravdu zapomněl lék užít, měl by se spíše poradit se svým lékařem.

Formát obsahu štítků jsem zvolil následující (první případ je pro první modul obvyčejného rozeznání, druhý případ se týká druhého modulu pro evidenci léků):

- *X|poznámka*, kde *X* představuje pořadové číslo štítku, *roua* | jako povinný oddělovač a za ní následuje textová poznámka, ke které uživatel přiřazuje audio poznámku až při rozeznání daného kódu, více viz níže
- *XP|poznámka*, kde *X* představuje pořadové číslo štítku, *P* jako označení pro léky, zbylé elementy jsou totožné s předchozím případem

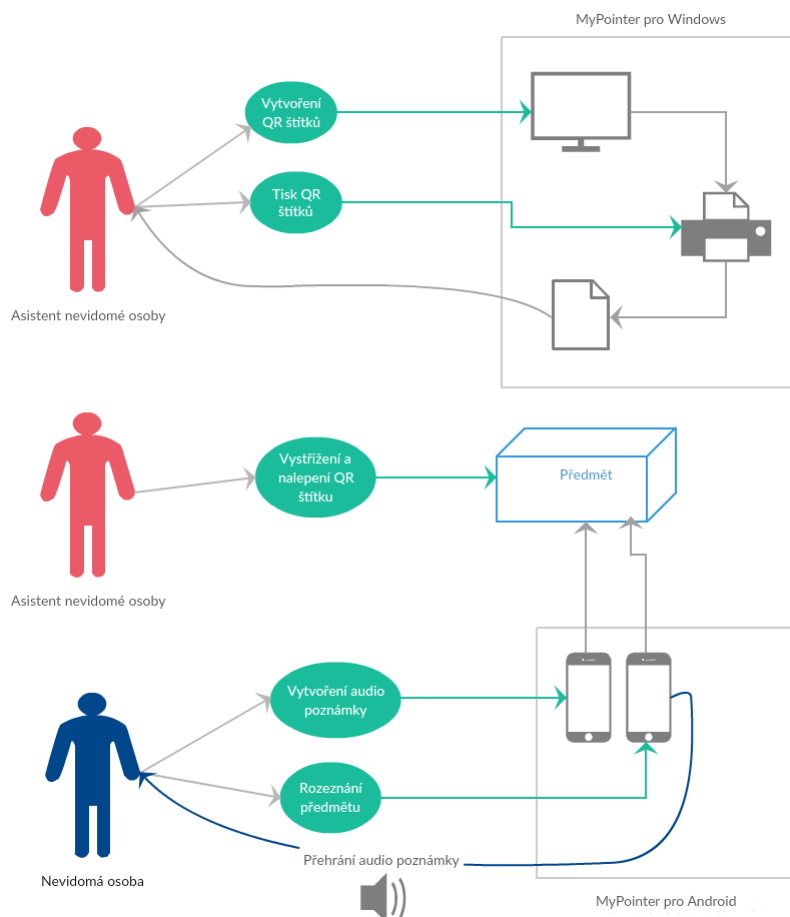
Zmíněná poznámka má více způsobů využití. První se týká již vytváření štítků v aplikaci pro operační systém Windows, kdy uživatel v seznamu přidaných štítků jasně vidí, jaký název štítku již do tohoto seznamu vložil, případně jaký štítek může odebrat.

Druhé využití se týká Android aplikace, kdy je využita Android třída *TextToSpeech*, která dokáže předčítat syntetizovaný text [13]. V případě, že by nahraná audio poznámka nebyla v úložišti dostupná, dojde k přehraní této poznámky pomocí zmíněné *TextToSpeech* třídy.

Tato poznámka je také využita jako nadpis v hlavičce obrazovky po rozeznání štítku. To může být využito v případě, kdy by uživatel nebyl zcela nevidomý, nebo pokud by poznámky ke štítkům nahrávala třetí osoba (bez zrakových potíží). Tento nadpis by mohl být také využit, pokud by si k určitému štítku chtěl nahrát audio poznámku jiný uživatel s vlastním mobilním zařízením, kde by byla aplikace MyPointer také nainstalována.

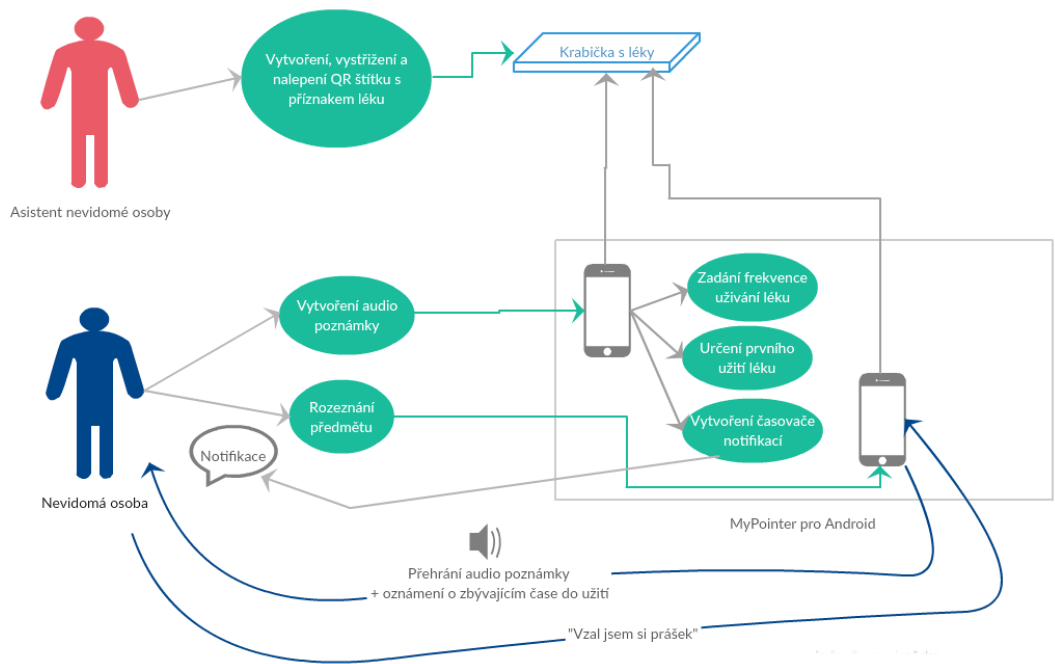
Aplikace pro operační systém Windows bude uživateli umožňovat definovat QR štítky a vytisknout je na tiskárně. Takové štítky dokáže aplikace zpracovat a nahrát k nim poté audio poznámky.

Pro znázornění funkcí aplikací jsou dále uvedeny následující use-case diagramy. Čtení QR štítků (viz Obrázek 4.1) probíhá tak, že nejprve osoba nápomocná nevidomé osobě vytvoří a vytiskne tyto QR štítky pomocí aplikace vytvořené pro Windows. Následně je vystříhne a nalepí na požadované předměty, aby k nim byly následně pomocí Android aplikace přidány audio poznámky. Tyto nahrané poznámky jsou při dalším rozeznání daného štítku Android aplikací přečteny.



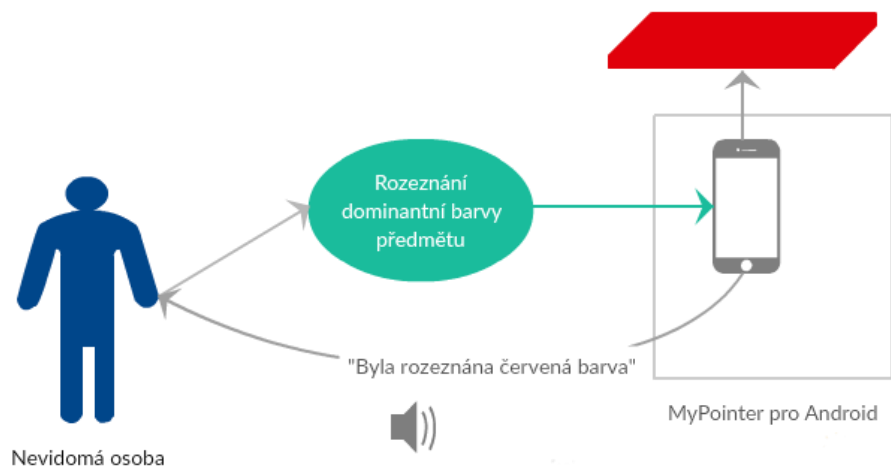
**Obrázek 4.1: Use-case diagram čtení QR štítku**

U čtení QR štítků týkajících se léků probíhá jejich vytvoření obdobným způsobem, jen je třeba je ve Windows aplikaci MyPointer náležitě označit zaškrtnutím pole s popiskem „Jedná se o kód s lékem“. Po naskenování uživatel nahraje k příslušnému štítku audio poznámku, zadá frekvenci užívání v hodinách a zvolí, za kolik hodin chce lék poprvé užít. Následně je vytvořen časovač pro zaslání notifikací, které jsou uživateli zobrazeny v době užití léku v notifikační části mobilního zařízení. Léky mohou být označeny jako užité, pokud zbývá do jeho užití jedna hodina. Tento use-case je znázorněn na Obrázku 4.2.



**Obrázek 4.2: Use-case diagram u štítků s léky**

Rozeznání dominantní barvy probíhá tak, že uživatel po zvolení třetího modulu z hlavního menu Android aplikace vyfotí pomocí svého mobilního zařízení fotografii daného předmětu a rozeznaná barva je mu následně sdělena po zpracování fotografie pomocí hlasového průvodce, více viz Obrázek 4.3.



**Obrázek 4.3: Use-case rozeznání dominantní barvy**

## 5. Použité technologie

V této kapitole jsou popsány technologie a knihovny, které jsem během implementace aplikace použil. Je nastíněn postup vývoje a zmíněny důvody použití či nepoužití zmíněných rozhraní a knihoven.

### 5.1. Android

Android je open-source softwarový balík obsahující operační systém, middleware<sup>7</sup> a klíčové aplikace spolu s API knihovnami umožňující vytváření mobilních aplikací [14]. Je vyvíjena společnostmi Google a Open Handset Alliance a je vydávána s otevřeným zdrojovým kódem. V současné době (datováno k roku 2016) je tento operační systém dostupný v 70 jazycích a vydání jeho poslední verze Android N proběhlo 11. dubna 2016, avšak nejedná se zatím o plně stabilní verzi.

K vývoji aplikací pro platformu Android se používá balík nástrojů s názvem Android software development kit (Android SDK). Programovacím jazykem pro Android je Java [15], která může být doplněna například o C a C++. Pro vývoj lze použít vývojové prostředí Android Studio vydané v roce 2014 společností Google, které je založené na vývojovém prostředí IntelliJ IDEA, jenž jsem použil z důvodu předešlé zkušenosti a trvající podpory tohoto prostředí ze strany společnosti Google. Pro multiplatformní vývoj (Android, iOS, Windows) lze použít například nástroj Xamarin od stejnojmenné společnosti, která byla v únoru roku 2016 koupena společností Microsoft. Ta Xamarin poté uvolnila zdarma pro vývojáře používající jakoukoliv edici vývojového prostředí Visual Studio.

### 5.2. QR kód

QR kód, z angl. Quick Response Code, je prostředek pro automatizovaný sběr dat a pro počítačové zpracování informací, který byl vytvořen v Japonsku. Jeho

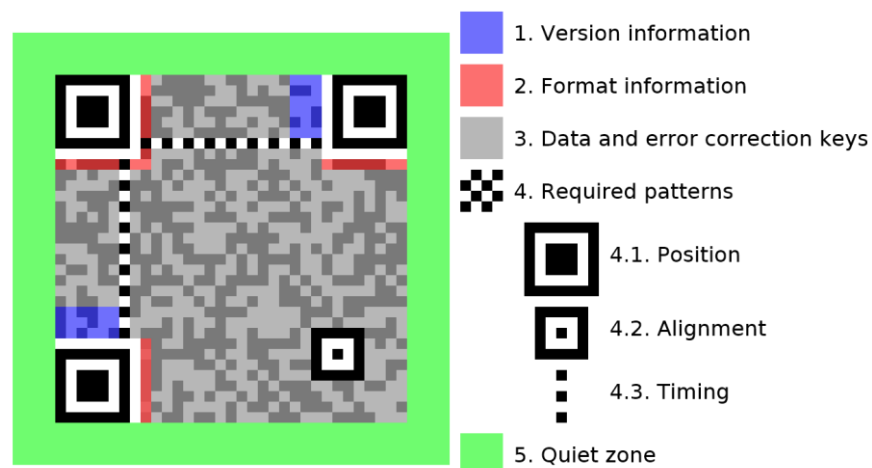
---

<sup>7</sup> Specializovaný software, který poskytuje aplikacím služby nad rámec služeb poskytovaných operačním systémem



specifikace je standardizována od roku 2000 pod ISO 18004 a tento standard byl upraven v roce 2006 [16]. Data obsažená v tomto kódu mohou být čitelná i při částečném znečištění, či poškození. Může obsahovat písmena, číslice, binární data nebo japonské znaky *kandži* a je schopný pojmout různé množství informací podle obsahu, maximálně však 7089 číslic, 4296 alfanumerických znaků, 2953 binárních osmibitových dat a 1817 *kandži* znaků.

Tento kód je definovaný ve čtyřiceti různých velikostech, kód verze  $V$  je tvořen čtvercovou mřížkou bodů rozměru  $17 + 4*V$ . Co se týče kapacity, může obsahovat až 4 296 písmen nebo číslic. Struktura kódu je znázorněna na Obrázku 5.1. Jsou zde informace o verzi kódu (1), informace o formátu kódu (2), data a samoopravné kódy (3), povinné vzory (4) a tzv. tichá zóna.



**Obrázek 5.1: Struktura QR kódu**

QR kódy se dají číst při jakémkoliv natočení. Oproti běžnému čárovému kódu má také QR kód nízkou cenu tisku díky menší velikosti, a to přibližně o jednu desetinu. Vzhledem k tomu, že štítky načítané mou aplikací mají být tisknutelné a lepené na předměty, byl toto hlavní důvod, proč jsem tento druh kódu upřednostnil před běžným čárovým kódem.

### 5.2.1. Knihovna Zxing

Pro vytvořenou aplikaci jsem ke čtení QR kódu použil knihovnu Zxing, jejíž

jádro umožňuje čtení, ale i vytváření čárových a QR kódů. Zxing („zebra crossing“) je open-source<sup>8</sup> multiplatformní knihovna zpracovávající obrázky čárových kódů, která je dostupná v několika portacích, například pro Python, Ruby, .NET, C++, ale i Android. Její portace pro Android umožňuje detekovat takový kód pomocí fotoaparátu a následně rozkódovat jeho obsah.

Zvolený formát obsahu šítka je blíže popsán v kapitole 4.3. Návrh aplikací.

### 5.3. OpenCV knihovna

Pro modul sloužící k rozpoznání barev jsem se nejprve rozhodl použít knihovnu OpenCV (Open Source Computer Vision), což je multiplatformní knihovna sloužící ke zpracování obrazu v reálném čase [17]. Je původně vyvíjena společností Intel. Hlavní důvod jejího vývoje bylo poskytnout společnou infrastrukturu pro aplikace z odvětví počítačového vidění a pro urychlení strojového vnímání. Knihovna je dostupná volně k užití jako open-source software.

Knihovna je vytvořena nativně v jazyce C++, existují však i její verze pro jazyky C, Python, Java a MATLAB. Podporuje operační systémy Windows, Linux, Android a MacOS. Obsahuje více než 2500 optimalizovaných algoritmů sestávajících jak z klasických algoritmů, tak z algoritmů užívaných ve vědních oborech, jako je například strojové učení.

Pro vývoj na platformě Android slouží speciální balík OpenCV4Android SDK, který se musí importovat do daného projektu a poté je kompilován spolu s aplikací. Po prvotních problémech s importem jsem zjistil, že velikost aplikace vzrostla přibližně o 40 MB a doba jejího nasazení a spuštění v mobilním zařízení se prodloužila zhruba o minutu. Tím se stalo testování a ladění aplikace mnohem obtížnějším.

Z této knihovny jsem používal funkce pro konverzi obrázku z barevného schématu RGB do schématu HSV a dále pro vypočtení histogramu. Snažil jsem se postupovat dle dostupné dokumentace OpenCV určené pro vývoj v jazyce Java, avšak jsem zjistil, že je v aktuální verzi (3.1.0) většinou stručná a pro vývojáře neznalé této

---

<sup>8</sup> Počítačový software s otevřeným zdrojovým kódem

knihovny těžko použitelná. V této fázi jsem také narážel na pády aplikace, které jsem nebyl schopen vyřešit ani s pomocí analýzy logového záznamu, který neobsahoval žádné nosné informace, o které by se dalo opřít.

Vzhledem k těmto nepříjemnostem a také při povážení, že prakticky by aplikace oproti jiným zabírala spoustu místa, rozhodl jsem se zvolit jiný postup a upustit od této knihovny, která byla pro mé potřeby příliš komplexní.

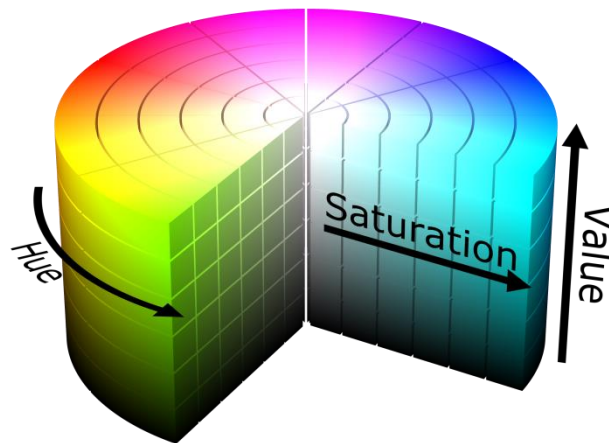
## 5.4. ColorThief knihovna

Tato knihovna je zaměřena na zjištění barevné palety obrázku a jeho dominantní barvy. K tomu využívá programovací jazyk JavaScript a HTML značku *canvas*, což je značka využívaná právě pro kreslení v prohlížeči pomocí JavaScriptu. Kromě zjištění dominantních barev se dá tato knihovna prakticky využít ještě například pro přizpůsobení uživatelského rozhraní dle zobrazovaného obsahu.

Pro svou aplikaci jsem použil portaci této knihovny pro jazyk Java, jejíž autor uvádí, že výpočet je oproti té originální až čtyřicetkrát rychlejší [18]. Na rozdíl od předchozí OpenCV knihovny (39 MB) zabírá ColorThief přibližně 0,5 MB.

Jelikož však tato knihovna pracuje s *Java AWT* třídami, které nejsou pro Android podporovány, bylo zapotřebí tuto knihovnu upravit tak, aby dokázala pracovat s Android grafickými knihovnami *android.graphics*. Konkrétně šlo o záměnu původní *BufferedImage* třídy s třídou *Bitmap*, což je třída podporovaná Androidem, která se nachází v balíku *android.graphics* [19].

Pomocí knihovny ColorThief je zjištěno barevné složení ve formátu odpovídajícímu schématu RGB (red, green, blue). Toto schéma je založeno na aditivním míchání barev, tedy způsobu, kdy se jednotlivé složky barev sčítají a vytváří světlo větší intenzity. Výsledná intenzita se rovná součtu intenzit jednotlivých složek. Každá z těchto tří barev je udána svou mohutností, přesněji počtem bitů pro barevnou komponentu, kdy pro 8 bitů na komponentu je rozsah hodnot pro mohutnost 0 – 255.



**Obrázek 5.2: HSV barevný model**

Daná dominantní barva reprezentovaná v RGB barevném modelu je poté převedena do HSV modelu. To proto, že tento model lépe separuje odstín (H – hue) od ostatních složek, na rozdíl od RGB modelu, kdy na danou rozeznanou barvu má značný vliv stín. HSV je barevný model, který sestává ze tří složek (viz Obrázek 5.2) – hue (barevný tón), saturation (sytnost barvy) a value (hodnota jasu), který více odpovídá lidskému vnímání barvy [20]. Obecně se zmíněný barevný tón označuje názvem barvy. Sytnost barvy představuje množství šedi v poměru k odstínu a měří se v procentech od 0% do 100%. Hodnota jasu neboli množství bílého světla v podstatě vyjadřuje, kolik černé barvy je do základní barvy přidáno.



**Obrázek 5.3: Rozložení barev ve složce hue**

Rozeznání barvy je určeno pomocí složky hue (odstín) a jejího následného přiřazení dle hodnoty  $0^{\circ}$ - $360^{\circ}$ . Pro přiřazení barvy jsem použil výčet zvolených základních barev, pro které jsem určil interval rozsahu odstínů s přihlédnutím k jejich rozložení (dle obrázku 5.3) následovně:

- Červená ( $0^{\circ}$ - $29^{\circ}$ ,  $330^{\circ}$  -  $360^{\circ}$ )
- Oranžová ( $30^{\circ}$ - $44^{\circ}$ )

- Žlutá (45°-89°)
- Zelená (90°-149°)
- Azurová (150°-179°)
- Modrá (180°-269°)
- Fialová (270°-299°)
- Růžová (300°-329°)

Z této složky odstínu by však nebylo možné rozeznat bílou a černou barvu. Bílé a černé barvě byly metodou testování určeny prahové hodnoty pro detekci. Pro bílou barvu byla tato prahová hodnota určena ve složce sytosti, pro černou pak ve složce jasu, více viz kapitola 8.

## 5.5. SQLite databáze

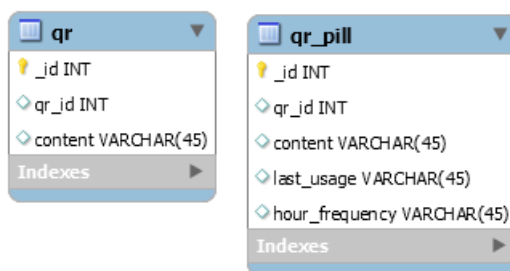
Pro ukládání strukturovaných dat v Android aplikacích je nejvhodnější použití databáze. Součástí jádra vývojového prostředí pro operační systém Android je SQLite knihovna [21]. Aplikace poté přistupují k SQLite pomocí rozhraní v balíčku *android.database.sqlite*.

SQLite je knihovna, která implementuje samostatný, transakční SQL databázový engine, který nepotřebuje vlastní server a nepotřebuje být jakkoliv konfigurován. Její autoři se rozhodli tuto knihovnu distribuovat jako volné dílo, tudíž může být využita pro jakýkoliv účel, ať už komerční, tak soukromý.

Při kompilování projektu se tato databáze přilinkuje k aplikaci a uloží se do souboru s příponou *.dbm* do adresáře */databases* uvnitř dané aplikace. Výhodou zde může být multiplatformovost, čili tento soubor můžeme libovolně přenášet mezi systémy. Nevýhodou zde může naopak být fragmentace. Co se týče datových typů, SQLite je od ostatních databází odlišná tím, že nepoužívá statické typování, ale dynamické. U statického typování je typ hodnoty určen definicí daného sloupce. Ve SQLite můžeme sice také sloupce definovat, ale i přesto je poté možné uložit do daného sloupce hodnotu jiného typu. Ve své aplikaci jsem využil dvě tabulky (viz Obrázek 5.4):

- qr – tabulka obsahující data týkající se QR kódů

- qr\_pill – tabulka obsahující data týkající se QR kódů s léky



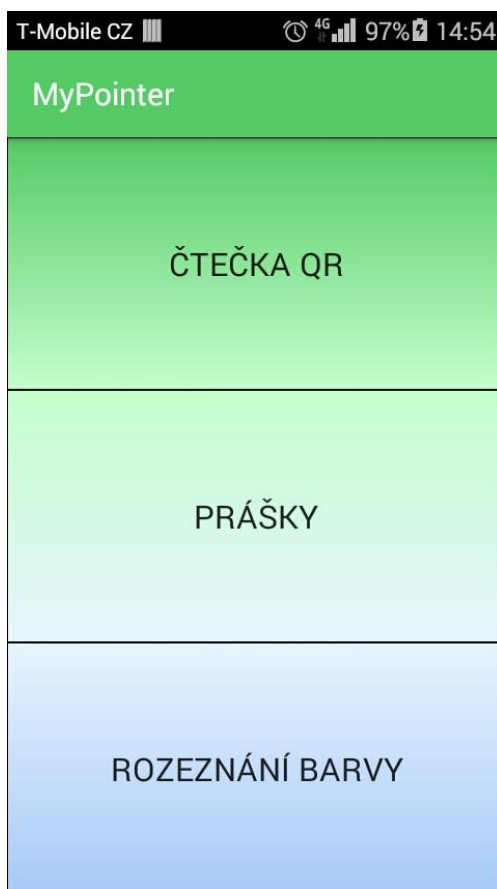
**Obrázek 5.4: ERA model Android aplikace**

## 5.6. Grafické uživatelské rozhraní

Grafickým uživatelským rozhraním Android aplikace rozumíme vše, co uživatel na obrazovce mobilního zařízení vidí a s čím může provádět určitou interakci. Veškeré prvky uživatelského rozhraní jsou v aplikaci Android tvořeny za použitím takzvaných View (pohledů), které jsou sdružovány do skupin pomocí ViewGroup (soubor pohledů). View je objekt, který se zobrazuje uživateli na obrazovce a ten s ním pak může dále interagovat. Android dále poskytuje podtřídy těchto View, které nabízí mnoho ovládacích prvků a modelů rozvržení (dále *layout*). Grafické uživatelské rozhraní je pro každou komponentu aplikace definováno použitím hierarchie složené z View a ViewGroup objektů, kde každá ViewGroup představuje neviditelný kontejner udržující potomky View, které pak mohou následně zobrazovat jednotlivé prvky na obrazovce.

Vytvoření layoutu je možné pomocí vytvoření instance View přímo ve zdrojovém kódu, avšak je zde také možnost jej definovat pomocí XML souboru. Jména XML elementů v tomto souboru odpovídají Android třídám. Ve své aplikaci jsem v layoutech nejčastěji používal LinearLayout, což je typ layoutu, který uspořádává své potomky do jedné buňky, či řádku. Orientace tohoto uspořádání může být definována jako horizontální, či vertikální. Problematika layoutů je dále zpracována v [22]. Další nejčastější komponentou užívanou v layoutu mé aplikace byla tlačítka (Button), kterými jsem díky vertikální orientaci LinearLayoutu graficky rozdělil menu aplikace na tři výše zmíněné moduly (viz Obrázek 5.5). Jelikož spolu dva první moduly souvisí co se QR kódů týče a abych je oddělil od modulu třetího, podbarvil jsem je zeleným nádechem,

třetí modul na rozeznání barvy jsem pak podbarvil modře.



**Obrázek 5.5: Rozdělení aplikace MyPointer na tři moduly**

Jak již bylo zmíněno v kapitole 4.2, aplikace je vhodná pro použití nevidomými lidmi. Co se týče ovládání mobilních zařízení s dotykovou obrazovkou nevidomými lidmi, na těchto obrazovkách nejsou žádné pomocné výstupky, které jsou například na počítačových klávesnicích (klávesa "F" a "J") a které bývaly na mobilních zařízeních s tlačítky. Díky nim mohli tyto lidé zařízení lépe ovládat pomocí hmatu. Aby bylo možno aplikaci ovládat poslepu, bylo zapotřebí grafické uživatelské prvky rozmístit tak, aby nedocházelo k nechtěným akcím. Maximálně jsou tedy na obrazovce rovnoměrně umístěny tři grafické prvky, které jsou navíc vždy pod sebou. Aby však uživatel věděl, do které části obrazovky má stisknout, je pomocí hlasového průvodce v aplikaci naváděn k ovládání.

Pokud by uživatel netrpěl úplnou ale částečnou vadou zraku, jsou veškeré popisky grafických uživatelských prvků zobrazeny velkým písmem.

## 5.7. Hlasový průvodce

Pro umožnění ovládání aplikace i osobami s úplnou zrakovou ztrátou, je uživatel celou aplikací naváděn hlasovým průvodcem, který radí, jak aktuální obrazovku použít. Tohoto průvodce jsem nahrál v domácím studiu do jednotlivých zvukových stop dle příkazů a začlenil do aplikace. Aby byla hlasová nápověda průvodce k uživateli svižná, snažil jsem se během nahrávání zvukových stop danou poznámku přečíst co nejrychleji, avšak v rámci srozumitelnosti.

Po nahrání stop jsem provedl postprodukcí, která obsahovala kompresi dynamiky audia, ekvalizaci a nakonec jsem aplikoval limiter. Kompresi dynamiky zvukového signálu je proces, díky kterému dojde ke zmenšení dynamického rozsahu daného zpracovávaného signálu – srovná drobné dynamické rozdíly, tedy že sníží část hlasu, která je hlasitější a zvýší tu část hlasu, která je tišeji. Ekvalizací lze dosáhnout úprav frekvenční charakteristiky zvukového signálu zesílením či potlačením některých částí akustického pásma. Vycházel jsem ze zkušenosti, že mobilní zařízení neumí produkovat nižší (basové) frekvence. Od toho se odvíjela ekvalizace a vyzdvihl jsem spíše středové frekvence. Nakonec jsem použil již zmíněný limiter, díky němuž dojde k nastavení kompresního poměru na maximum a to způsobí, že veškeré hlasové pasáže překračující práh (špičky) jsou vyrovnány na tento nastavený práh.

Aby si však tyto audio stopy zachovaly určitou kvalitu a zároveň kvůli nim nezabírala aplikace příliš místa, bylo potřeba exportovat nahrávky z *.wav* do jiného formátu. Android podporuje velké množství zvukových formátů [23], z nich nejvhodnější pro moji potřebu z nich byl formát *.ogg*, díky kterému byla velikost všech nahrávek snížena z 27 MB na 1,3 MB a kvalita zvuku byla přesto dostačující. V úvahu byl brán také formát *.mp3*, soubory však byly jednou tak velké v porovnání s *.ogg* a rozdíl kvality audia byl nepatrný.

Hlasový průvodce obsahuje nápovědu pro uživatele ohledně aktuální pozice v aplikaci a o možných postupech aplikací dále. Jsou zde také skládány řetězce s hodnotami, jako například informace o aktuální nastavené frekvenci hodin užívání léků, či informaci o názvu rozpoznané barvy.

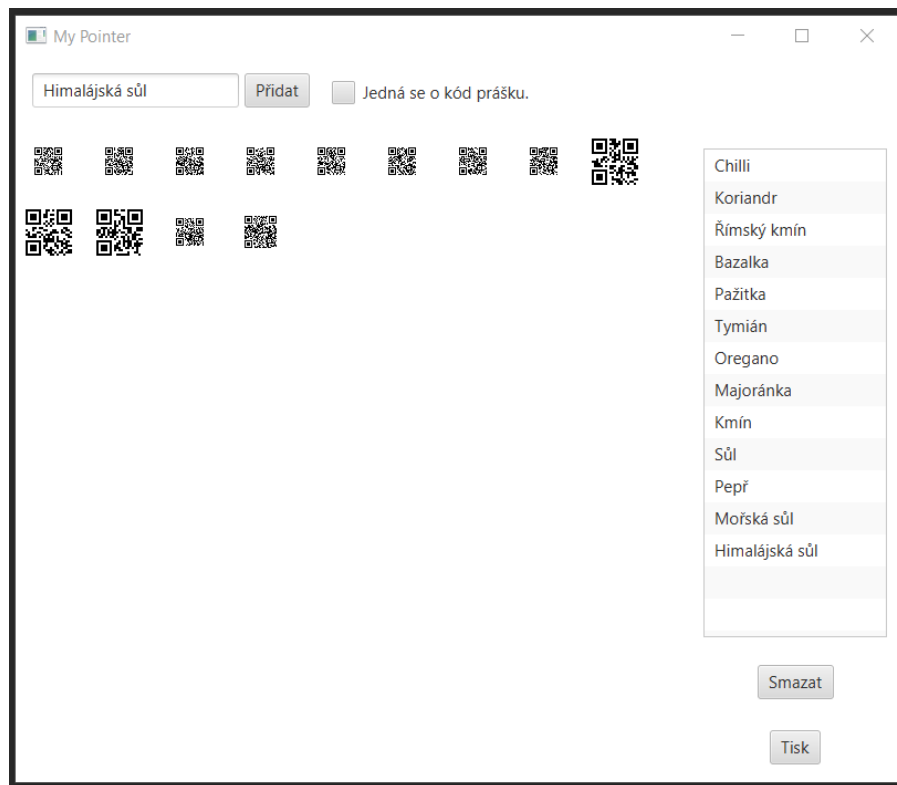


## 5.8. JavaFX

JavaFX je platforma, která je založena na bázi platformy Java, je vyvinuta společností Sun Microsystems a její distribuce je licencována jako open-source (s otevřeným zdrojovým kódem). Tato platforma byla poprvé představena v roce 2007 a aktuální stabilní verze 8 vyšla v březnu roku 2014, která zároveň oficiálně nahradila platformu Swing, která již neměla dostatečnou podporu nových technologických prvků, jako podporu dotykových obrazovek, schopnost vytvářet animace, atp.

JavaFX slouží primárně pro vývoj tzv. RIA aplikací (z angl. Rich Internet Applications), neboli webových aplikací [24]. Díky tomu umožňuje zajistit interaktivitu webových stránek. Díky této platformě je navíc možné danou aplikaci ze stránek stáhnout a poté ji používat jako desktop aplikaci. Pro tvorbu rozhraní aplikace se používá Scene Builder, což je nástroj, díky němuž lze jednotlivé prvky uživatelského rozhraní libovolně přetahovat do plochy aplikace. Z těchto prvků je pak na pozadí vygenerován tzv. FXML kód. Tyto prvky lze pomocí doplnění atributu *fx:id* prolinkovat do Java tříd pomocí anotace *@FXML* a následně je v Java třídě číst, či jinak dále zpracovávat. Více o JavaFX například v knize [25].

Aplikace byla vytvořena za účelem vytisknutí QR štítků na PC s operačním systémem Windows, se kterými pak bude Android aplikace schopna pracovat dále. Po vepsání poznámky ke štítku dojde k vygenerování QR kódu a přidání do kolekce (viz Obrázek 5.6), kterou je poté možno vytisknout pomocí dialogu pro tisk. Danou poznámku je po vložení také možno smazat, pokud by se uživatel omylem spletl a chtěl akci vrátit zpět. Maximálně je možno na papír velikosti A4 vytisknout 60 QR kódů. Tuto hranici jsem určil během experimentování s tiskem QR štítků, kdy jsem také určil rozměr takového štítku, a to 1,5 cm na 1,5 cm. Taková velikost mi přišla jako ideální pro nalepení na menší předměty, zároveň ale není kód poté tak malý, aby se s ním špatně manipulovalo při nalepování.



**Obrázek 5.6: JavaFX aplikace pro vytváření štítků**

## 6. Vývoj mobilní aplikace

Tato kapitola obsahuje popis tříd a metod vytvořené během vývoje aplikace. Jsou zde popsány jejich metody a u Android aplikace je popsána struktura projektu.

### 6.1. Struktura projektu v Android Studiu

Aplikaci jsem vyvíjel ve vývojovém prostředí Android Studio. Součástí instalace Android Studia je samotné Android Studio IDE<sup>9</sup>, Android SDK Tools (souhrn nástrojů pro vývoj a debugování aplikací), kompilátor Android a základní emulátory s plnohodnotným systémem Android. Pomocí Android Studia je možno kromě samotného vývoje například testovat kompatibilitu, podepisovat aplikace certifikátem a navrhovat grafické uživatelské rozhraní.

Android projekt je spjat s buildovacím nástrojem Gradle, což je buildovací nástroj, který umí projekt kompilovat, testovat, nasazovat a exportovat [26]. Je kombinací nástroje pro sestavování softwarových aplikací Ant a systému závislostí Maven, což je repozitář zkompilovaných knihoven, jejich dokumentace a někdy také jejich zdrojového kódu. Výhodou použití Gradlu je značné zrychlení sestavení aplikace díky tomu, že používané knihovny jsou již staženy zkompilované. Další výhodou je možnost specifikovat verze knihoven, na jejichž případné konflikty umí upozornit.

Struktura projektů je založena na jistých konvencích, které byly dodrženy v mé aplikaci. Zdrojové kódy jsou umístěny na cestě *src/main/java*, zdroje aplikace (resources) jsou uloženy na cestě *src/main/resources*. Veškerá konfigurace nástroje Gradle se zapisuje do souborů *.gradle*, kdy kořenový soubor *build.gradle* obsahuje pouze základní informace jako verze gradlu a používaný repozitář závislostí. V souboru *settings.gradle* je pak umístěn odkaz na moduly, které jsou v projektu obsaženy a které se mají kompilovat (v mém případě modul *app* obsahující samotnou aplikaci a modul *java-colorthief-master* obsahující knihovnu pro určení dominantní barvy z obrázku).

V souboru *build.gradle* v adresáři představující modul *app* jsou pak uvedeny

---

<sup>9</sup> Z angl. Integrated Development Environment – vývojové prostředí

hlavní informace o verzích aplikace (Android API), jako cílová, minimální a verze, proti které se aplikace bude kompilovat. Dále je zde ID aplikace (odpovídá názvu balíčku) a je zde také seznam závislostí, čili seznam zkompileovaných knihoven. Po takovém zkompileování je vytvořen soubor s příponou *.apk*, která obsahuje veškeré potřebné zdroje pro samostatný chod aplikace.

Během vývoje jsem využíval dokumentaci určenou přímo pro vývojáře Android aplikací od společnosti Google<sup>10</sup>. Některá řešení jsem také čerpal ze stránek StackOverflow, což je celosvětová online komunita sdružující programátory, kde mohou sdílet své zkušenosti a tázat se na řešení svých problémů<sup>11</sup>.

### **Struktura projektu aplikace:**

`/.gradle` – složka se staženými zkompileovanými knihovnami nástroje Gradle

`/.idea` – obsahuje konfigurační soubory vývojového prostředí Android Studio

`/app` – obsahuje zdrojový kód aplikace a zdrojové soubory (resources)

`/build` – adresář generovaný Android build systémem, který obsahuje sestavenou aplikaci ve formátu *.apk*

`/gradle` – obsahuje Gradle wrapper, jenž definuje, jaká verze nástroje Gradle je pro projekt použita a spolu s ostatními vygenerovanými soubory slouží k tomu, aby při přenesení projektu jinému vývojáři nemusela být opět stahována daná verze nástroje Gradle

`/java-colorthief-master` – projekt s knihovnou Java ColorThief pro rozpoznání dominantní barvy z obrázku

`build.gradle` – kořenový konfigurační soubor nástroje Gradle definující cestu k repozitáři Maven závislostí a verzi nástroje Gradle

`gradle.properties` – soubor umožňující nastavení vlastností spouštění aplikace, například konfigurace alokace paměti

---

<sup>10</sup> Dostupné z <https://developer.android.com/index.html>

<sup>11</sup> Dostupné z <https://stackoverflow.com>

`gradlew` – start-up skript pro UNIX systémy

`gradlew.bat` - dávkový soubor, pomocí kterého nástroj Gradle spouští sestavování aplikace

`local.properties` – soubor generovaný Android Studiem, kde je definovaná cesta k Android SDK

`MyPointer.iml` – soubor generovaný vývojovým prostředím, kde jsou definovány cesty ke knihovnám, pluginům a Android či Maven komponentám

`settings.gradle` – obsahuje odkaz na 2 moduly, které mají být nástrojem Gradle importovány a to modul `app` a knihovna `Java ColorThief`

## 6.2. Struktura Android aplikace

Aplikace se skládá z patnácti tříd a využívá dvě knihovny (`ColorThief` v modulu rozeznání barvy a `Zxing` v modulech pro čtení QR štítků)

### 6.2.1. Obecné třídy

V této podkapitole jsou veškeré třídy (aktivity), které dědí od třídy `AppCompatActivity`. Pod pojmem aktivita si lze představit základní komponentu pro vykreslení grafického návrhu aplikace.

Pro znázornění struktury aplikace je přiložen diagram tříd, viz Příloha E. Tento diagram je barevně rozdělen dle tří modulů aplikace.

#### A) `MyPointerMainActivity.java`

Inicializační třída (aktivita), která je užita jako vstupní bod aplikace. Při jejím načtení inicializuje a spouští hlasového průvodce (třídu `CustomMediaPlayer`), a nastavuje odkazy na všechny tři moduly aplikace.

### **Významné metody:**

```
onCreate(Bundle savedInstanceState)
```

– nastavuje grafické rozhraní aplikace, inicializuje tlačítka a přiřazuje jim odkazy na aktivity obsluhující jednotlivé moduly; pokud se jedná o opětovné vstoupení do menu, uvítací zpráva již není přehrána

### **B) QrReaderActivity.java**

Aktivita umožňující přečtení QR štítku, přehrání jeho poznámky, nahrání poznámky nové, smazání kódu a přenahrávání nové poznámky k existujícímu štítku.

### **Významné metody:**

```
initializeOnCreate()
```

- inicializuje připojení k databázi, inicializuje aktivitu pro skenování štítku

```
onActivityResult(int requestCode, int resultCode, Intent intent)
```

- pokud je návratová hodnota z aktivity obsluhující fotoaparát správná (úspěšně zparsována), testuje se QR kód dle regex výrazu – kód musí být ve tvaru:

*X/poznamka* – kde *X* je ID číslo, | je povinný oddělovač a za ním poznámka

- při rozeznání kódu přehraje audio poznámku, pokud je dostupná – jinak je pro přehrání použita třída Text-to-Speech, která přehraje danou poznámku z obsahu štítku v elementu *poznámka*

- při rozeznání nového kódu je dále uživateli umožněno nahrát poznámku novou

```
playQRNote(QRDao qrDao)
```

- přehraje audio poznámku, pokud je dostupná

- pokud dostupná není, použije se třída Text-to-Speech

```
handleNewQRCode(QrDao qrDao)
```

- přehraje poznámku o tom, že byl rozeznán nový kód a inicializuje tlačítko pro nahrání nové audio poznámky k příslušnému QR štítku

```
startRecording(Button btnRecStart, QRDao qrDao)
```

- metoda pro umožnění nahrání audio poznámky, kdy je tento audio soubor uložen do složky dat aplikace pod názvem souboru *id.3gp*, kde *id* je ID číslo QR štítku

```
maxRecDurationReached(QRDao qrDao)
```

- metoda pro odchyčení maximálního možného času pro nahrání poznámky, který byl experimentálně stanoven na čtyři sekundy; po uplynutí této doby se nahrávání zastaví

- tento čas lze dle potřeby změnit v hlavní třídě aplikace *MyPointerMainActivity*, kde je definován konstantou *MAX\_RECORD\_DURATION\_MS*, jenž představuje tento čas v milisekundách

```
handleRecordedNote(QRDao qrDao)
```

- umožňuje uživateli rozhodnout, zda chce danou audio poznámku uložit, přehrát či přenahrát novou

```
setStateBetweenScans(QRDao qrDao)
```

- inicializuje tlačítka pro přesměrování do menu, nahrání nové audio poznámky k předešlému kódu, či pro volbu nového skenování QR štítku

### C) QrPillReaderActivity.java

Aktivita umožňující přečtení QR štítku obsahující informace ohledně léku, jeho smazání či nahrání nové audio poznámky. Metody týkající se čtení QR štítků a nahrávání audio poznámek jsou shodné s předchozí třídou, pouze je zde upraven regex výraz:

*XP/poznamka* – kde přibyl příznak *P* pro označení léků, díky kterému se tento typ štítků ukládá do jiné databázové tabulky

Navíc jsou zde však metody pro určení frekvence užívání léků a vypočtení doby

následného užití léku. Více o této frekvenci v kapitole 4.3.

#### **D) CaptureActivityAnyOrientation.java**

Tato třída dědí od CaptureActivity a díky ní je pomocí konfigurace v souboru AndroidManifest.xml umožněno skenovat QR štítky i v horizontální poloze mobilního zařízení.

#### **E) ColorRecognizeActivity.java**

Aktivita, díky které dojde pomocí fotoaparátu k rozeznání dominantní barvy z vyfoceného obrázku.

##### **Významné metody:**

```
onCreate(Bundle savedInstanceState)
```

- tato metoda inicializuje grafické rozhraní aktivity a táže se zařízení, zda je fotoaparát dostupný

```
processBitmapImage(Bitmap image)
```

- metoda využívá knihovnu ColorThief pro zjištění dominantní barvy z obrázku zachyceného pomocí fotoaparátu mobilního zařízení

- zjistí HSV barevné schéma rozeznané dominantní barvy a dle algoritmu popsaného v kapitole 5.4 přiřadí dané barvě její název a odkaz na audio poznámku této barvy

- je zde však kontrola, zda složka Value nedosahuje hodnoty menší než 17%, pak je barva klasifikována jako černá

- tato kontrola je také pro bílou barvu, a to pokud je složka Saturation menší než 6%, je tato barva klasifikována jako bílá



## 6.2.2. Databázové třídy

Tato kapitola obsahuje popis tříd, které komunikují s databází SQLite v mobilním zařízení.

### A) DBHelper.java

Tato třída dědí od třídy DBHelper, díky níž je možno ovládat databázi SQLite a spravovat její verze. Touto třídou je obsluhována tabulka s názvem *qr*, která obsahuje veškeré informace o základních QR štítcích. Vytvoření této tabulky při první inicializaci databáze je umožněno přetížením metody `onCreate`, kdy dojde k provedení skriptu pro založení této tabulky.

#### Významné metody:

```
isQrStored(QRDao qrDao)
```

- vyhledá daný QR štítek v databázi dle ID čísla a pokud jej nalezne, je vrácena instance tohoto štítku, jinak je návratová hodnota *null*

```
saveQr(QRDao qrDao)
```

- uloží daný QR štítek do databáze

- v této metodě dochází také k rozeznání, zda se jedná o uložení nového QR štítku, či o přepsání stávajícího

### B) DBPillHelper.java

Tato třída má obdobnou funkci jako třída předchozí, avšak na rozdíl od ní spravuje štítky týkající se léků. Tabulka obsluhovaná touto třídou má název *qr\_pill*.

### C) QRDao.java

Jedná se o entitní třídu, která mapuje hodnoty QR štítků na objekty.

#### **D) QRPillDao.java**

Tato entitní třída mapuje hodnoty QR štítků týkajících se léků na objekty.

#### **E) QRTable.java**

Třída reprezentující QR štítek v databázi, neboli třída reprezentující její tabulku. Je zde umístěna vnitřní třída QRDaoEntry. Ta představuje jednotlivý záznam v tabulce. Tato vnitřní třída implementuje rozhraní *BaseColumns*, díky němuž jsou přidány sloupce ID řádku *\_ID* a počet řádků v tabulce. Tato hodnota je využívána při mazání a přidávání záznamů. V této vnitřní třídě jsou dále definice všech sloupců tabulky.

#### **F) QRPillTable.java**

Tato třída je obdobná té předchozí, avšak slouží pro reprezentaci QR štítku týkajícího se léků.

### **6.2.3. Pomocné třídy**

Tato podkapitola obsahuje popis všech pomocných tříd, které využívají aktivity aplikace.

#### **A) CustomMediaPlayer.java**

Třída obsluhující přehrávání audio poznámek QR štítků a hlasového průvodce. Umožňuje přehrávání více stop za sebou, díky řetězení nahrávek.

#### **B) FileUtility.java**

Třída obsluhující ukládání audio nahrávek QR štítků. Pomocí jejích metod lze

kopírovat audio stopy z dočasného adresáře poskytovaného operačním systémem Android do adresáře aplikace.

### **C) HoursFrequency.java**

Třída výčtového typu reprezentující hodiny. Slouží pro určení frekvence užívání léku a pro vypočtení doby prvního užití od vytvoření informace o léku.

### **D) MyColor.java**

Třída výčtového typu reprezentující zvolené barvy, které jsou aplikací rozeznávány. Každá barva obsahuje název, interval odstínu na stupnici 0°-360° a odkaz na její příslušnou audio nahrávku.

### **E) RemindService.java**

Tato třída slouží pro vytvoření časovače na notifikaci ohledně užití léku. Tento časovač se zaregistruje při vytváření audio poznámky ke QR štítku s lékem. Uživatel je pak pomocí notifikace upozorněn, že si má daný lék vzít. Časovač na tyto připomínky lze vypnout pouze odstraněním daného QR štítku.

## 7. Vývoj desktopové aplikace

Tato aplikace slouží pro vytisknutí QR štítků. Jedná se o JavaFX aplikaci. Vyvíjel jsem ji v prostředí IntelliJ IDEA, jež tuto platformu podporuje, včetně Scene Builderu, což je prostředí pro tvorbu grafického rozhraní. Aplikace se skládá ze dvou tříd a jedné *FXML* třídy definující grafické rozhraní. Je zde využita knihovna QRGen pro generování QR kódů.

### 7.1. Main.java

Třída dědicí od abstraktní třídy *Application*, díky jejíž přetížené metodě *start* lze k aplikaci přilinkovat soubor *mypointer.fxml* definující grafické rozhraní.

### 7.2. Controller.java

Třída, která implementuje od rozhraní *Initializable*, díky čemuž lze inicializovat uživatelské prvky, jako seznam štítků a panel pro umístění těchto štítků pro následný tisk.

#### Významné metody:

```
print(Node node)
```

- díky této metodě dojde ke změně rozměrů, aby bylo možno štítky vytisknout na papír ve formátu A4

- dále je zobrazeno okno pro výběr tiskárny a následný tisk

```
addRow()
```

- metoda přidá danou poznámku do seznamu

- pokud dojde k naplnění listu, který je omezen na 60 QR kódů, uživatel je upozorněn vyskakovacím oknem

```
deleteRow()
```

- smaže vybraný kód označený v seznamu z listu QR kódů

## 8. Testování

Aplikace byly testovány průběžně během vývoje a poté byly podrobeny uživatelským testům. Z těchto testů byly sepsány připomínky, na jejichž základě došlo k následným úpravám.

Pro testování jsem nejčastěji využíval výpisu do logu ve vývojovém prostředí Android Studio, pomocí kterého lze například filtrovat určité slovo, či ihned změnit úroveň logování i při běhu aplikace. Druhou nejčastější metodou testování bylo krokování (debugování) aplikace. Nevýhodou testování Android aplikací je stále doba jejich nasazení a spuštění na mobilním zařízení, či emulátoru.

Při testování JavaFX jsem také využíval logování a debugování, oproti testování Android aplikace zde probíhalo sestavení a spuštění řádově několik sekund.

### 8.1. Testování Android aplikace

Pro rozhodnutí o způsobu testování jsem se nejprve ujistil, zda nedošlo od mé předchozí zkušenosti s testováním v emulátoru k nějakým změnám ohledně doby jeho startu a následného spuštění aplikace. Tato doba je však stále nepříznivá, a proto jsem se rozhodl pro testování pomocí fyzického mobilního zařízení připojeného přes USB port.

Testování probíhalo na mém smartphonu Huawei Ascend Y550, ve kterém je operační systém Android verze 4.4.4, RAM paměť o velikosti 1,0 GB a úhlopříčka jeho obrazovky je 4,5". Na tomto zařízení proběhlo veškeré testování a byly odladěny nalezené chyby. Aplikace zde fungovala plynule.

Zprvu jsem ladil pouze drobné chyby týkající se špatného prolinkování grafických prvků z uživatelského rozhraní do kódu.

Při práci se Zxing knihovnou, která slouží pro zpracování QR kódů, jsem narazil na problém, kdy bylo mobilní zařízení během skenování kódu otočeno o devadesát stupňů (do tzv. landscape módu), během čehož došlo k pádu aplikace. Tento problém jsem vyřešil implementací třídy `CaptureActivityAnyOrientation`, která dědí od `CaptureActivity`, a díky konfiguraci `android:screenOrientation="fullSensor"` v souboru

obsahujícím základní informace o aplikaci (AndroidManifest.xml) byla tato chyba opravena.

Druhý problém nastal při vývoji hlasového průvodce. Android k účelu přehrávání audia nabízí třídu Media Player, která však neumožňuje snadné přehrávání více stop po sobě. Tento problém jsem potřeboval vyřešit z toho důvodu, že hlasový průvodce v některých částech aplikace skládá hodnoty za sebou, jako například „Aktuální hodnota“ + „pět hodin“. Pro tento účel jsem implementoval pomocnou třídu CustomMediaPlayer, která obsahuje kolekci (list) objektů Integer, kde každá položka kolekce reprezentuje odkaz na zdroj (resource) dané přehrávané stopy. V metodě pro přehrávání je pak nastaven listener, díky němuž dojde k přehrání další stopy v této kolekci.

Během testování bylo zapotřebí odchytné také chyby způsobené nečekanými vstupy od uživatele. Jedním z takových případů bylo možné nahrání QR kódů s informací, kterou by aplikace nedokázala zpracovat. Abych tedy určil formát vstupu a abych dokázal rozlišit, zda se jedná o obyčejný QR štítek nebo štítek s léky, definoval jsem dva formáty, dle kterých jsou štítky vytvářeny (blíže viz. kapitola 4.3). Pro dodržení těchto formátů je obsah štítků kontrolován regex výrazem.

## **8.2. Testování JavaFX aplikace**

Při testování této aplikace docházelo k problémům, kdy se na tiskovou plochu nevešly všechny přidané štítky. Proto jsem určil maximální počet štítků, které je možno do mřížky přidat.

Narazil jsem zde na chybu, kdy docházelo k přehrání poznámky ke štítku v Android aplikaci poznámkou novou. Aby tomu tak nebylo, je index každého štítku zvyšován, následně ukládán do souboru a při opětovném spuštění aplikace načten.

## **8.3. Uživatelské testy**

V následující kapitole jsou popsány jednotlivé uživatelské testy, které byly realizovány v poslední fázi vývoje Android aplikace.

### 8.3.1. Modul pro rozeznání barvy

Předpokladem tohoto testu je použití aplikace na mobilním zařízení, které má fotoaparát. Zároveň je potřeba, aby daný předmět, jehož barvu chceme zjistit, byl držen v dostatečně blízké vzdálenosti a mezi tímto předmětem a mobilním zařízením nebyly žádné překážky (prst na čočce, dlaň, jiný předmět). Pro správné určení barvy tohoto předmětu je potřeba, aby zabíral na obraze většinu plochy.

- Zapněte aplikaci MyPointer na svém mobilním zařízení
- Uslyšíte úvodní přivítání a hlasový průvodce oznámí informaci, že pro modul rozeznání barvy je potřeba stisknout dolů
  - Stiskněte tlačítko v dolní třetině obrazovky
- Zapne se aktivita pro vyfotografování obrázku
  - Zaměřte mobilní zařízení na daný předmět a vyfoťte jej
- Uslyšíte zprávu o tom, jaká barva byla rozeznána

#### Výsledky

Test proběhl úspěšně a podařilo se rozeznat danou barvu. Tento test jsem zkoušel vícekrát a došel jsem k závěru, že mnou zvolená paleta barev je rozeznávána vcelku spolehlivě, avšak u bílé a černé muselo dojít ke korekcím prahových hodnot kvalifikace.

Pokud se však daná barva více lišila od zvoleného rozsahu barev, aplikace rozeznávala barvy odlišného odstínu. Toto chybné rozeznávání se také objevovalo při horších světelných podmínkách, zejména při nedostatku světla při fotografování. Vliv na klasifikaci má také kvalita fotoaparátu mobilního zařízení. Jelikož fotoaparát na mém zařízení není příliš kvalitní, už na pořízené fotografii jsem zaznamenal, že barva není přesně taková, jaká vypadá na pohled.

Jako jedno z možných rozšíření tohoto modulu by bylo vhodné přidat více barev, které by však nebyly rozeznávány pouze pomocí složky *hue*, ale musely by zde být brány v potaz všechny tři složky, díky kterým by se pomocí nastavení prahových hodnot dané barvy rozeznávaly.

### 8.3.2. Modul pro čtení QR štítků

Stejně jako u následujícího testu je u těchto dvou modulů pracujících s QR štítky zapotřebí, aby byla nevidomé osobě poskytnuta pomoc s použitím JavaFX aplikací a s vytisknutím a vystřížením QR štítků. V tomto případě není možné jiné řešení. Dále se předpokládá, že uživatel má k dispozici funkční tiskárnu pro černobílý tisk na papír ve formátu A4.

V následujícím testu je vyzkoušeno použití aplikace MyPointer s QR štítky nalepenými na kořenkách.

- Zapnutí aplikace MyPointer pro Windows
- Zobrazí se hlavní okno aplikace - vytiskněte QR štítek s názvem koření:
  - Zadejte název koření
  - Potvrďte stisknutím tlačítka Přidat
  - Stiskněte tlačítko Tisk a vyberte příslušnou tiskárnu pro tisk
- Vytiskne se stránka s QR štítkem
  - Vystříhnete QR štítek
  - Nalepte jej na kořenku
- Zapněte Android aplikaci MyPointer
- Uslyšíte úvodní přivítání a hlasový průvodce oznámí informaci, že pro modul čtení QR štítků je potřeba stisknout nahoru
  - Stiskněte do první třetiny obrazovky
- Zobrazí se aktivita pro načtení QR kódu pomocí fotoaparátu
  - Namiřte mobilním zařízením na QR kód
- Dojde k rozeznání nového kódu, uživatel je informován o nahrání nové poznámky ke kódu
  - Stiskněte na obrazovku a vyslovte poznámku
- Po čtyřech sekundách se nahrávání stopne, uložte poznámku
  - Stiskněte do horní třetiny obrazovky
- Naskenujte znovu daný kód
  - Stiskněte do horní třetiny obrazovky a zamiřte fotoaparátem na kód
- Dojde k přečtení nahrané poznámky



## Výsledky

Test proběhl úspěšně a došlo k přehrání dané audio poznámky. Při tomto testu jsem narazil na potíže, kdy knihovna Zxing déle rozeznávala QR kód pomocí fotoaparátu, avšak tato doba nepřesáhla pět sekund.

### 8.3.3. Modul s léky

U tohoto testu je proces vytisknutí a nalepení štítků obdobný s testem předchozím. Daný QR štítek může být nalepen na krabičku od léků.

- Zapněte aplikaci MyPointer pro Windows
- Zobrazí se hlavní okno aplikace - vytiskněte QR štítek s názvem léku:
  - Zadejte název léku
  - Zaškrtněte pole „Jedná se o kód s lékem.“
  - Potvrďte stisknutím tlačítka Přidat
  - Stiskněte tlačítko Tisk a vyberte příslušnou tiskárnu pro tisk
- Vytiskne se stránka s QR štítkem
  - Vystříhnete QR štítek
  - Nalepte jej na krabičku od léku
- Zapněte Android aplikaci MyPointer
- Uslyšíte úvodní přivítání a hlasový průvodce oznámí informaci, že pro modul s léky je potřeba stisknout doprostřed
  - Stiskněte doprostřed
- Zobrazí se aktivita pro načtení QR kódu pomocí fotoaparátu
  - Namiřte mobilním zařízením na QR kódu
- Dojde k rozeznání nového kódu, uživatel je informován o nahrání nové poznámky ke kódu
  - Stiskněte na obrazovku a vyslovte poznámku
- Po čtyřech sekundách se nahrávání stopne, uložte poznámku
  - Stiskněte do horní třetiny obrazovky
- Zobrazí se aktivita pro vybrání frekvence užívání léku – vyberte dvě hodiny
  - Stiskněte pětkrát do horní třetiny obrazovky pro zvolení dvou

hodin

- Uložte stisknutím uprostřed
- Zobrazí se aktivita pro určení prvního užití – vyberte možnost ihned
  - Stiskněte dlouze uprostřed
- Po uplynutí doby 2 hodin naskenujte znovu daný kód a označte lék jako „užitý“
  - Naskenujte znovu tento QR kód a stiskněte do horní poloviny obrazovky

### **Výsledky**

Test proběhl úspěšně, došlo k přehrání informace o zbývajícím času do užití léku. Uživatel má zde možnost označit lék jako právě užitý, poté dojde k přepočítání doby do dalšího užití dle nastavené frekvence.

## 9. Rozšíření aplikace

Aplikace je navržena tak, aby mohla být případně dále rozšířena. Rozšíření by se mohlo týkat například přidáním dalšího druhu štítků, které budou rozeznávány. Tato funkce by byla realizována tak, že by byl přidán jiný příznak do obsahu štítku, než je písmeno *P* u QR štítků s léky, rozšířena databáze a převzaty funkce pro zpracování QR kódu a audio poznámky.

Další rozšíření by se mohlo týkat notifikací, které by mohly být případně pokročilejším způsobem kontrolovány z aplikace při rozeznání daného štítku, jelikož nyní jsou notifikace pouze vytvářeny a mazány spolu s vytvořením a smazáním štítku.

Notifikace by mohly být také v podobě zaslání SMS jiné osobě, což by mohlo být využito u seniorů v případě, že by si zapomněli vzít lék a nevšimli si upozornění. SMS by byla zaslána buď rodinným příslušníkům, sousedům či personálu domova pro seniory.

Modul pro rozeznání dominantní barvy by mohl být později rozšířen o více barev ke klasifikaci. Toto rozšíření je již naznačeno na konci kapitoly 8.3.1. Tento modul by také mohl být rozšířen o detekci textu, což by využili uživatelé neschopní přečíst malý text, avšak tato funkce by se odvíjela také od kvality fotoaparátu mobilního zařízení. Nevidomí by tuto funkci nejspíše nebyli schopni využít, pokud by nejdříve nebyli poučeni o pozici tohoto textu.

## 10. Závěr

Cílem této práce bylo zmapovat možnosti operačního systému Android pro pomoc lidem s různými zdravotními obtížemi. Nejprve bylo zapotřebí provést analýzu těchto možností, jak už v podobě vestavěných funkcí systému Android, tak i dosud existujících aplikací. Dalším z cílů bylo vybrat vhodnou funkcionalitu pro usnadnění běžných každodenních činností lidí s vybraným handicapem. Tato funkcionalita byla vybrána a realizována, vytvořená aplikace byla otestována. Byly stanoveny cíle práce, konkrétně vytvoření aplikace nápomocné lidem se zrakovými potížemi.

Implementoval jsem JavaFX aplikaci spustitelnou na operačním systému Windows, která umožní uživateli vytisknout dva druhy QR kódů. Jedním takovým typem může být běžné použití (například označení kořenek, přihrádek s hřebíky), druhým typem je označení obalů od léků. Tuto aplikaci jsem testoval a opravil objevené chyby.

Dále jsem implementoval Android aplikaci, která sestává ze tří modulů. První dva moduly dokáží již zmíněné štítky dále zpracovat. Uživatel k nim díky aplikaci přiřadí hlasovou poznámku, která je při příštím rozeznání QR kódu přehrána. V případě léků lze navíc nastavit frekvenci užívání léku a uživatel tak může mít přehled, zda si již daný lék vzal, či nikoliv. Třetí modul dokáže rozeznat dominantní barvu z obrázku, který byl pořízen uživatelem pomocí fotoaparátu daného mobilního zařízení a tuto barvu nahlas přečíst pomocí hlasového výstupu. Tento hlasový výstup zároveň napomáhá uživateli orientovat se v aplikaci. Byl kladen důraz na snadné ovládání a snadno rozeznatelné prvky grafického uživatelského rozhraní.

Android aplikace byla testována na fyzickém mobilním zařízení. Bylo objeveno několik chyb, které byly následně opraveny. Chod aplikace je plynulý a věřím, že by mohla pomoci lidem se zrakovými obtížemi v některých běžných každodenních činnostech.

Jako možné rozšíření bylo navrženo přidat více barev do vybrané palety pro klasifikaci. Tyto barvy by však bylo třeba klasifikovat pomocí všech tří HSV složek a určit jejich prahové hodnoty. U modulu pro připomínání léků by mohlo být přidáno pokročilejší nastavení připomínání.

# Literatura

1. *Výběrové šetření zdravotně postižených osob - 2013* [online]. 2014 [cit. 2016-04-12]. Dostupné z: <https://www.czso.cz/csu/czso/vyberove-setreni-zdravotne-postizenych-osob-2013-qacmwuvwsb>
2. *Asistivní technologie* [online]. [cit. 2016-04-12]. ISSN 1804-6517. Dostupné z: [http://www.wikiskripta.eu/index.php/Asistivn%C3%AD\\_technologie](http://www.wikiskripta.eu/index.php/Asistivn%C3%AD_technologie)
3. *Rozdělení zdravotně postižených osob do skupin*. In: *Žiju s handicapem* [online]. 2012 [cit. 2016-05-09]. Dostupné z: <http://www.zijushandicapem.cz/clanky/zamestnani/rozdeleni-zdravotne-postizenych-osob-do-skupin.html>
4. *Sluchové postižení* [online]. [cit. 2016-04-12]. ISSN 1802-5145. Dostupné z: <http://www.helpnet.cz/sluchove-postizeni>
5. SERVUSOVÁ, Jana. *Kontrastivní lingvistika - český jazyk x český znakový jazyk*. 2., opr. vyd. Praha: Česká komora tlumočnicků znakového jazyka, c2008. ISBN 978-80-87218-30-3.
6. *Všeobecná encyklopedie v osmi svazcích*. Praha: Diderot, 1999. Encyklopedie Diderot. ISBN 80-902555-5-8.
7. *Introduction to Mental Retardation. The Arc*. ARC of the United States, Silver Spring, MD., 2004, : 2.
8. MATĚJČEK, CSC, Prof. PhDr. Zdeněk. *Sfumato - Splývavé čtení* [online]. In: . [cit. 2016-05-09]. Dostupné z: <http://www.sfumato.cz/metodika-sfumato.html>
9. KEJLOVÁ, Sabine. *Pracovní příležitosti hendikepovaných* [online]. 2013 [cit. 2015-12-28]. Dostupné z: <http://minority.fsv.cuni.cz/?p=384>
10. ŠKUBALOVÁ, Petra. *Zaměstnávání osob se zdravotním postižením*. Praha, 2012. Bakalářská práce. Vysoká škola ekonomická v Praze. Vedoucí práce Doc. Ing. Otakar Němec, CSc.
11. MKN-10: *Mezinárodní statistická klasifikace nemocí a přidružených zdravotních problémů* [online]. [cit. 2016-02-01]. Dostupné z: <http://www.uzis.cz/cz/mkn/index.html>
12. *Be My Eyes: Lend your eyes to the blind* [online]. [cit. 2015-12-28]. Dostupné z: <http://www.bemyeyes.org/>
13. *TextToSpeech: Class Overview*. Android Developers [online]. 2016 [cit. 2016-05-

- 09]. Dostupné z:  
<https://developer.android.com/reference/android/speech/tts/TextToSpeech.html>
14. MEIER, Reto. *Professional Android application development*. Indianapolis, IN: Wiley, 2009. ISBN 978-047-0344-712.
  15. MURPHY, Mark L. *The busy coder's guide to Android development*. 2nd ed. United States: CommonsWare, 2009. ISBN 9780981678009.
  16. ISO/IEC 18004:2006 [online]. [cit. 2016-04-21]. Dostupné z:  
[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=43655](http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=43655)
  17. MUHAMMAD, Amgad. *OpenCV Android Programming By Example*. Packt Publishing, 2015. ISBN 978-1-78528-293-5.
  18. *Color Thief - A Fast Java Implementation*. GitHub [online]. [cit. 2016-05-09]. Dostupné z: <https://github.com/SvenWoltmann/color-thief-java>
  19. *Bitmap*. Android Developers [online]. 2016 [cit. 2016-05-09]. Dostupné z: <https://developer.android.com/reference/android/graphics/Bitmap.html>
  20. PLATANIOTIS, Konstantinos N. a A. N. VENETSANOPOULOS. *Color image processing and applications*. New York: Springer, c2000. Digital signal processing (Springer-Verlag). ISBN 3540669531.
  21. ADITYA, S. K., KARN, V. K. *Android SQLite Essentials*. Packt Publishing, 2014. ISBN 9781783282951.
  22. KURNIAWAN, Budi. *Java for Android*. Second Edition. Brainy Software Inc, 2015. ISBN 9781771970259.
  23. MEDNIEKS, Zigurd, Laird DORNIN, G. Blake MEIKE a Masumi NAKAMURA. *Programming Android Download*. First Edition. O'Reilly Media, Inc, 2011. ISBN 978-1-449-38969-7.
  24. DEA, Carl, Mark HECKLER, Jose PEREDA a Sean PHILLIPS. *JavaFX 8: Introduction by Example*. 2nd ED. Apress, 2014. ISBN 978-1-4302-6460-6.
  25. ANDERSON, Gail a Paul ANDERSON. *Essential JavaFX*. Upper Saddle River, NJ: PRENTICE HALL, 2009. ISBN 0137042795.
  26. BERGLUND, Tim. a Matthew MCCULLOUGH. *Building and testing with Gradle*. Sebastopol, CA: O'Reilly Media, Inc., 2011. ISBN 9781449304638.

## **Slovník zkratek**

AAC - Augmentativní a alternativní komunikace

API – Application Programming Interface, rozhraní pro programování aplikací

AWT – Abstract Window Toolkit, knihovna grafických uživatelských prvků pro platformu Java

GPS – Global Positioning System, globální polohovací systém

IDE – Integrated Development Environment, integrované vývojové prostředí

iOS – Operační systém společnosti Apple

IT – Informační technologie

NLUI – Natural Language User Interface, uživatelské rozhraní přirozeného jazyka

OS – Operační systém

PC – Personal Computer, osobní počítač

QR – Quick Response, rychlá odpověď

SDK – Software Development Kit, sada vývojových nástrojů

TTS – Text-To-Speech, převod textu na řeč

USB – Universal Serial Bus, univerzální sériová sběrnice

XML – Extensible Markup Language, rozšiřitelný značkovací jazyk

# Seznam příloh

Příloha A – Rozhovor s třídní učitelkou

Příloha B – Dopis organizacím

Příloha C – Odpovědi na dopis

Příloha D – Uživatelská dokumentace

Příloha E – Diagram tříd



## **Příloha A: Rozhovor s třídní učitelkou**

Rozhovor s třídní učitelkou – speciálním pedagogem ze ZŠ speciální v Blatné, který proběhl v prosinci roku 2015.

*„Můžete mi prosím blíže charakterizovat skupinu dětí, se kterou ve třídě pracujete?“*

„Máme ve třídě šest dětí, z toho je pět v přípravném stupni a jeden imobilní chlapec v posledním ročníku školní docházky. Děti jsou ve věku 6-7 let, přípravný stupeň navštěvují třetím rokem. Jsou to většinou děti z rodin, pouze jedna dívka je z ústavní péče. Děti pocházejí z nepodnětného rodinného prostředí, zaostávají v celkovém psychomotorickém vývoji, dvě mají Downův syndrom, jeden chlapec trpí poruchou AUTI spektra a epilepsií. Všechny děti k nám přišly s vážnou poruchou komunikace, takže náš největší cíl byl, naučit je používat k dorozumění řeč.“

*„Jak probíhá výuka takto postižených dětí a na co se vlastně postupně zaměřujete?“*

„První rok je většinou zaměřen na sebeobsahu. Děti přijdou inkontinentní, nevládají samostatně stolovat, chovat se ve skupině dětí, nejsou zvyklé sedět u stolku a splňovat pracovní činnosti. Opravdu záleží na rodině, ve které vyrůstají. Ve druhém roce se pokračuje v upevňování získaných dovedností a návyků a věnujeme se více jemné a hrubé motorice a rozvoji smyslů. Třetí rok je předpřípravné období pro vstup na plnění povinné školní docházky. Každé dítě má vypracován svůj individuální vzdělávací program, který je dítěti zpracován na míru, čili se vzdělává podle jeho možností a schopností; čili podle toho se věnujeme grafomotorickému cvičení, učí se znát některé hlásky a číslice; dochází průběžně k socializaci – např. navštěvujeme různá zařízení ve městě, úřady, obchody, atd. Jinak rozvrh hodin je obsažen v dopoledním vyučování a to v předmětech – rozumová a smyslová výchova, pracovní a výtvarná, hudební a rehabilitační tělesná a samozřejmě řečová výchova.“

*„Jaké metody a pomůcky při výuce používáte?“*

Je pravda, že využíváme při výuce – kromě těch klasických - spoustu alternativních metod; záleží také na složení dětí ve třídě podle míry a druhu postižení. Vedle ve třídě pracují kolegyně s dorozumívací metodou VOKS, což je výměnný

komunikační obrázkový systém. A to u dětí, které se vůbec nenaučily používat hlas. U nás ve třídě se nově druhým rokem pracuje s metodou SFUMATO, což je nová metoda výuky čtení, kterou jsem se se svou kolegyní – asistentkou ve třídě rozhodla experimentálně vyzkoušet a ověřit si, zda by metoda nemohla pomoci dětem k rozvoji řeči. Používáme samozřejmě ještě mnoho dalších alternativních metod – Dobrý start, Začít spolu, Tvořivá škola - , ale musím podotknout, že veškeré pomůcky pro děti k výuce, ať už se jedná o kteroukoliv jmenovanou metodu, že jsme si museli vyrobit vždy sami. Tak např. ke SFUMATU – jsou to písmena nalepená na tvrdém kartonu, vystříhaná z barevné pryže, nebo zalaminovaná, či nalepená na větších dřevěných kostkách. Truhláři jsem vysvětlila, k čemu je budu potřebovat, jak asi velké, no a zbytek už byl na mně. Kostky se používají k syntéze hlásky, je zde důležitý zvuk – dítě vidí souhlásku a dlouze načítá samohlásku a při ťuknutí kostek do sebe by mělo dojít k syntéze a dítě vysloví hlásku.“

*„A osvědčila se metoda SFUMATO k rozvoji řeči u vašich žáků?“*

„Samozřejmě pracujeme současně podle kritérií logopedie, což se mi zdálo, že pro současnou strukturu dětí je velmi málo. Takže ano, lze jednoznačně odpovědět, že podle dostupných výsledků a šetření, které probíhalo během dvou školních let v naší přípravné třídě, lze tuto metodu čtení použít i k rozvoji řeči dětí v přípravném stupni ZŠ speciální. Je třeba však mít na mysli, že metoda byla plně přizpůsobena a v mnohých prvcích poupravena vzhledem k postižení dětí, jejich zvláštnostem a potřebám, protože práce na základní škole speciální je velmi odlišná od práce na ZŠ se zdravými dětmi. Vždy jsem vycházela z podstaty SFUMATA, že OKO – se vykmitává, nahlíží dostatečně dlouho na písmeno; HLAS – dlouze exponuje hlásku správně fyziologicky, učí se tzv. volat ve své mluvní tónině a SLUCH – se učí zvukovou podobu předložené litery exponované hlasem. Určitě by nám práci zjednodušila nějaká aplikace, program,...; který by bylo možno využít pro výuku na interaktivní tabuli, nebo na dotykovém monitoru PC.“

*„Děkuji vám za rozhovor a přeji mnoho úspěchů ve vaší práci.“*

„I já děkuji vám a znovu vás srdečně zvu na návštěvu do naší školy, třeba by vás něco napadlo, co by nám výuku k rozvoji řeči u našich dětí nějak usnadnilo. Ať se vám také daří.“

Rozhovor poskytla Mgr. Petra Klecánová.

## **Příloha B: Dopis organizacím**

*Dobrý den,*

*jmenuji se Štěpán Tuháček, studuji obor Informační systémy na ZČU a píši diplomovou práci na téma Android pro handicapované. Jelikož je mé téma poměrně široce zaměřené, rád bych toho využil, aby mohla má výsledná aplikace najít reálné využití a stala se případně takovýmto lidem nějakým způsobem nápomocnou.*

*Pokud byste našel/la volnou chvíli k zodpovězení mých otázek, budu Vám vděčný. Pokud by Vás zároveň napadl někdo, kdo by mohl přijít s nějakým dobrým nápadem, přepošlete, prosím, tento dopis dále.*

- a) Zajímalo by mě, zda znáte (příp. používáte) nějaké aplikace, která dokáže pomoci jakkoliv handicapovaným lidem?*
- b) Pokud ano, jak se jmenují, na jaké jsou platformě (Android, iOS, Windows) a chybí Vám v nich nějaká funkce, či napadá Vás nějaké možné vylepšení?*
- c) Chybí Vám na trhu nějaká aplikace, která by mohla pomoci handicapovaným lidem? Případně napadá Vás nějaká funkce, kterou by mohli takoví lidé využít?*

*Předem děkuji za Vaši odpověď. Pokud byste měl/a více nápadů, mohli bychom je případně vyřešit osobně.*

*Přeji hezký den,*

*Štěpán Tuháček*

## Příloha C: Odpovědi na dopis

*Hezký den,*

*mobilní aplikace nepoužíváme, proto Vám na zbývající otázky nedokážu odpovědět.*

*S přáním příjemného dne zdraví*

*Ing. Miroslava Grožajová*

*Vedoucí Centra pro lidi se zdravotním postižením, p.o.*

---

*Dobrý den,*

*Ráda vám na dotazy odpovím, i když nevím, zda vám tím pomohu. Pracujeme s iPad 4 v rámci sociální služby raná péče a to nejčastěji s dětmi s poruchou autistického spektra. Využíváme především aplikace o komunikaci a sociálním chování, relaxační aplikace a pak také aplikace rozvíjející logiku a pozornost. Vyberu vždy jednoho nejčastějšího zástupce z dané kategorie:*

- *Emotions – aplikace, kde dítě vybírá ze čtyř obrázků, dle uvedené emoce*
- *Series 1 (a další pokračování) – dítě má najít logiku mezi obrázky, např. stárnutí člověka od novorozeněte až po seniora*
- *Klábosil – komunikační aplikace, kde dítě, pokud z nějakého důvodu nemluví, komunikuje přes tuto aplikaci*
- *Fluidity – relaxační aplikace, kde dítě dotykem obrazovky mění barvy (připomíná mi to souhvězdí)*
- *Poté různé aplikace, kde si lze s „Talking cat“ čistit zuby, mýt si vlasy, apod., děti jsou na základě toho, že některý úkon provedou v aplikaci, zopakovat úkon na sobě*

*U většiny aplikací je největším problémem angličtina, se kterou si generace rodičů stále neumí poradit. Některé aplikace lze přemluvit, ale neumožňuje to každý iOS, což nás vede k tomu, že neaktualizujeme. Aplikace by dle mého také měly být uživatelsky jednodušší, a to jak pro rodiče, tak pro děti. Např. Klábosil je úžasná aplikace, ale uživatelsky velmi náročná na zorientování se. Aplikací chybí zajisté několik, především právě na českém trhu, ale nejsem schopna to více specifikovat, většinou jsem vždy našla,*

*co jsem potřebovala a pro své potřeby upravila.*

*Pokud budete mít další dotazy, ráda vám je zodpovím.*

*Mějte se fajn a držím palce při získávání dat,*

*Mgr. Markéta Berešová, Dis.*

*vedoucí Rané péče, MOTÝL o.s.*

*Žlutická 2, 323 00, Plzeň*

---

*Dobrý den,*

*ráda bych odpověděla, ale uvítala bych nějaký dotazník, případně i na internetu, abych na něj mohla lépe odpovědět. Domnívám se, že i v rámci DP by to bylo žádoucí než takto položené otázky.*

*S přáním hezkého dne*

*G. Smečková*

---

*Dobrý den,*

*ráda čtu, že se studenti zabývají aplikacemi pro lidi s postižením. Snad jim budou ku prospěchu. Sama mám starý telefon a v aplikacích se tudíž nevyznám, ale Váš email jsem přeposlala kolegyním, které by snad mohly dát nějaký tip. Napadá mě snad jen nějaké propojení s registrem poskytovatelů sociálních služeb:*

*[http://iregistr.mpsv.cz/socreg/rozsirene\\_hledani\\_sluzby.fw.do?SUBSESSION\\_ID=1443708315666\\_3](http://iregistr.mpsv.cz/socreg/rozsirene_hledani_sluzby.fw.do?SUBSESSION_ID=1443708315666_3)*

*Nebo v situaci, kdy si člověk s mentálním postižením neví rady, tak kontakt na nějakou organizaci (např. na naše poradenské centrum).*

*Přeju hodně zdaru!*

*Michaela Stoklasová*

*Poradenské centrum SPMP ČR*

*Společnost pro podporu lidí s mentálním postižením v České republice, z.s.*

*Karlínské náměstí 59/12, 18603 Praha 8*

# Příloha D: Uživatelská dokumentace

V této kapitole je popsáno, jak se s danými aplikacemi zachází a co vše uživatel nabízí.

## D.1: Android aplikace

### Instalace

Pro umožnění instalace aplikace MyPointer do mobilního zařízení je nejprve nutno povolit funkci „Neznámé zdroje“ v systémovém nastavení daného mobilního zařízení. Soubor *MyPointer.apk* uložíme do úložiště zařízení, a to buď do interního či externího (paměťová karta). Tento soubor se nachází na přiloženém CD v adresáři */Android\_apk*. Pomocí průzkumníka poté aplikaci nainstalujeme. Aplikaci je možno nainstalovat na mobilní zařízení s verzí systému Android 4.1 Jelly Bean a vyšší.

### Spuštění aplikace

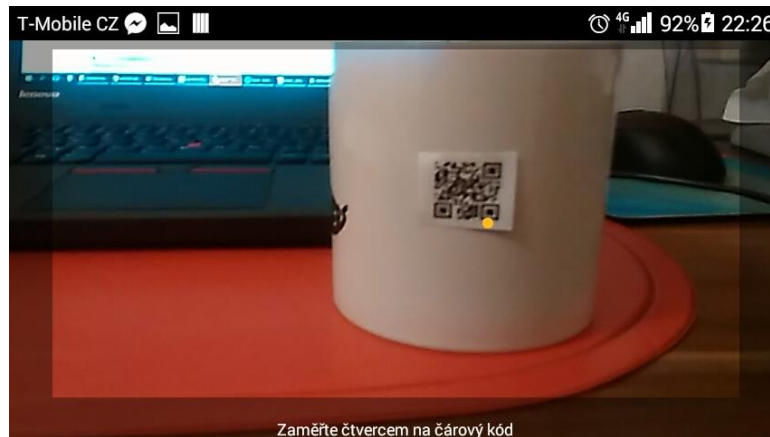
Po spuštění aplikace se zobrazí hlavní menu aplikace. Pro pomoc nevidomým uživatelům je zde hlasový průvodce, který pomáhá orientovat se v aplikaci. Z tohoto menu lze přistoupit ke třem modulům aplikace:

- Čtení QR štítků
- Připomínání léků
- Rozeznání barvy

### Čtení QR štítků

Pro zvolení prvního modulu stiskněte do horní třetiny obrazovky. Spustí se aktivita pro skenování QR štítků, namiřte tedy fotoaparát na daný štítek (viz Obrázek D.1).





**Obrázek D.1: Skenování QR štítku**

Po načtení štítku budete vyzváni k nahrání audio poznámky, kliknutím na obrazovku tak můžete učinit. Nahrávání lze zastavit opětovným stisknutím na obrazovku, nebo se toto nahrávání zastaví po čtyřech sekundách samovolně. Tento kód lze na následující obrazovce uložit stisknutím do horní třetiny obrazovky, přehrát stisknutím doprostřed, či nahrát poznámku novou, a to stisknutím do dolní třetiny obrazovky.

Po uložení dané audio poznámky lze provést nové skenování stisknutím do horní třetiny obrazovky, nahrát k poslednímu naskenovanému štítku novou audio poznámku stisknutím doprostřed, nebo se vrátit se do menu stisknutím do dolní třetiny obrazovky.

Následným opětovným naskenováním tohoto uloženého štítku dojde k přečtení uložené audio poznámky.

### **Připomínání léků**

Z menu lze stisknutím doprostřed vybrat modul pro připomínání léků. Jako u předchozího modulu se následně zobrazí aktivita pro naskenování QR kódu. Následuje nahrání audio poznámky, tento postup je opět shodný s modulem předešlým. Po uložení audio poznámky budete vyzváni ke zvolení frekvence užívání léku. Tato frekvence určuje, po kolika hodinách chcete lék užívat. Frekvenci nastavíte pomocí horních (+) a dolních (-) tlačítek, tlačítkem uprostřed ji uložíte.

Následně budete vyzváni, abyste zadali dobu, kdy chcete určit první užití léku. Tuto dobu lze opět nastavit v hodinách jako v předchozím případě, dlouhým stisknutím doprostřed lze však nastavit, že jste si lék vzali v tuto chvíli.

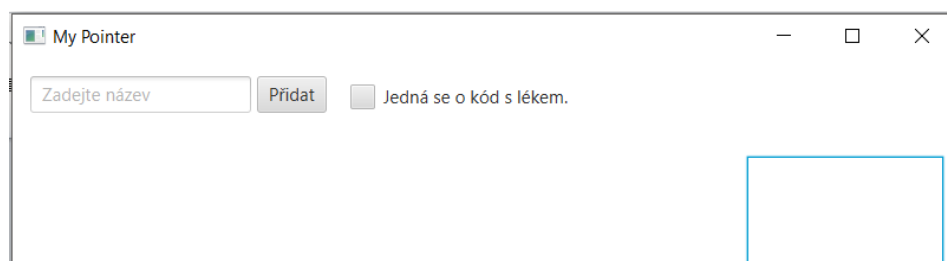
Po opětovném naskenování štítku s tímto lékem dojde k přečtení nahrané audio poznámky a budete informováni, za kolik hodin si máte lék vzít. Po uplynutí této doby můžete označit lék jako „užitý“ a dojde k přepočítání doby pro užití léku dle zadané frekvence užívání.

### Rozeznání barvy

Stisknutím do dolní třetiny v menu vyberete modul pro rozeznání dominantní barvy z vyfocené obrázku. Následně se zobrazí aktivita pro vyfocení fotografie. Namířte mobilní zařízení na zkoumaný objekt (měl by být dostatečně blízko, aby jeho barva převládala mezi ostatními) a stiskněte spoušť fotoaparátu. Následně dojde ke zpracování fotografie a k přečtení rozeznané barvy. Poté budete přesměrováni zpět do menu aplikace.

## D.2: JavaFX aplikace

Po spuštění JavaFX aplikace, která se nachází na přiloženém CD ve složce */JavaFX*, bude zobrazeno okno pro vytisknutí QR štítků na papír ve formátu A4 (viz Obrázek D.2).



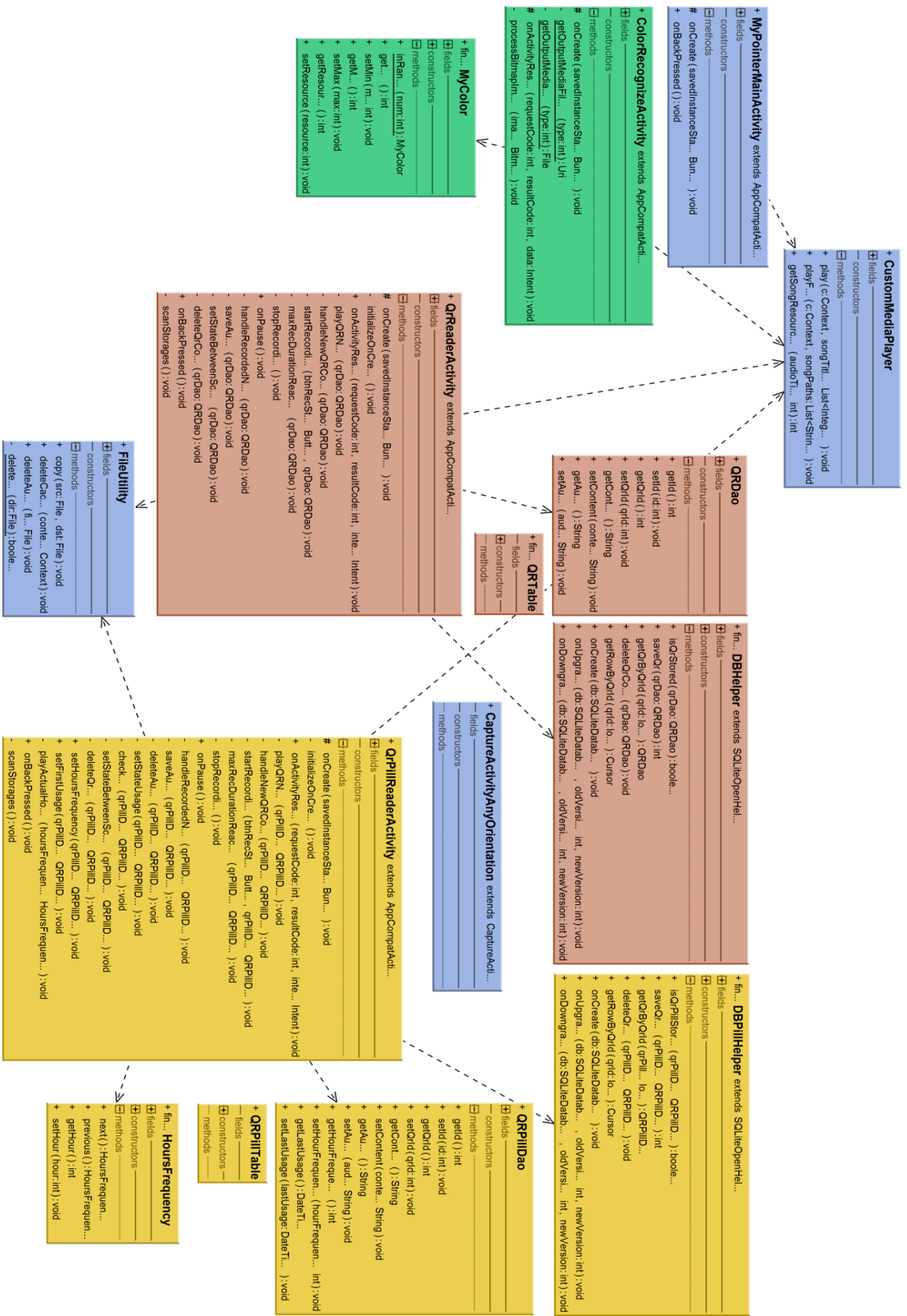
**Obrázek D.2: JavaFX aplikace MyPointer pro vytvoření QR štítků**

Nyní můžete zadávat názvy štítků do textového pole a tlačítkem „Přidat“ je vložíte do seznamu napravo. Zároveň se vygenerují QR kódy, které je poté možno

tlačítkem „Tisk“ vytisknout na černobílé tiskárně na papír ve formátu A4. Z tohoto seznamu lze štítky také odebírat tlačítkem „Smazat“.

Tyto štítky pak vystřihněte a nalepte na předměty, jenž budete následně rozpoznávat pomocí Android aplikace MyPointer.

# Příloha E: Diagram tříd



# Obsah CD

Příložené CD má následující adresářovou strukturu:

- **app**: Instalační soubor s android aplikací (*.apk*) a JavaFX aplikací (*.jar*)
- **attachment**: Přílohy diplomové práce
- **doc**: Zdrojový i tisknutelný text diplomové práce
- **javadoc**: Vygenerovaná JavaDoc dokumentace
- **src**: Zdrojové kódy Android aplikace i JavaFX aplikace