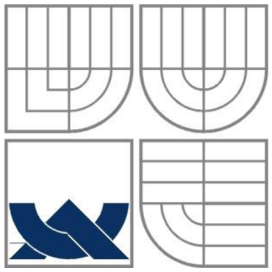
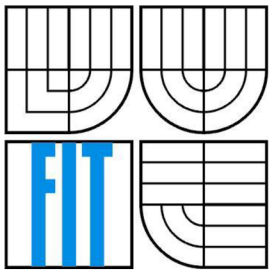


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

CONTEXT-AWARE FILTR NOTIFIKACÍ PRO ANDROID

CONTEXT-AWARE NOTIFICATION FILTER FOR ANDROID

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
SAMUEL JAKLOVSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR
ING. ISTVÁN SZENTANDRÁSI

BRNO 2016

Zadání bakalářské práce/18621/2015/xjaklo02

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Jaklovský Samuel**
Obor: Informační technologie
Téma: **Context-aware filtr notifikací pro Android**
Context-Aware Notification Filter for Android
Kategorie: Uživatelská rozhraní

Pokyny:

1. Prostudujte existující možnosti context-aware filtrace notifikací.
2. Seznamte se s Android SDK.
3. Navrhněte aplikaci na zadávání filtrů a systém na zjištění stavu a prostředí ze senzorů.
4. Implementujte aplikaci pro uživatelské testování.
5. Proveďte rozsáhlé uživatelské testování.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte video pro prezentaci projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

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

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

Vedoucí: **Szentandrási István, Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2015
Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této práce je vytvořit aplikaci pro zařízení s operačním systémem Android, která na základe získaného kontextu určí uživatelský profil a následně aplikuje zvukové nastavení, které pro tento profil uživatel předem definoval. V práci je popsaná teorie a návrh uživatelského rozhraní, které bylo implementováno jako funkční aplikace. Aplikace využívá pro určování uživatelských profilů Naïve Bayes klasifikátor a rozhodovací strom. Funkcionalita aplikace byla úspěšně otestována dvaceti uživateli. Hodnocení v dotaznících se v průměru pohybovalo kolem osm a půl bodu z deseti maximálně možných. Tyhle výsledky je možno považovat za úspěšné.

Abstract

The goal of this thesis is to develop an application for devices running Android which will determine user profile, based on obtained context, and apply user pre-defined sound settings for this profile. The thesis contains a description of common theory and design of user interface which was implemented as fully operational application. The application uses Naïve Bayes classifier and Decision tree for determining the user profile. The functionality of the application was successfully tested by twenty users. The average ratings in the questionnaires were about eight and a half points from a possible maximum of ten. These results can be considered successful.

Klíčová slova

Android, kontext, kontext-awareness, filtr, notifikace, Bayes, grafické uživatelské rozhraní

Keywords

Android, context, context-awareness, filter, notifications, Bayes, graphical user interface

Citace

JAKLOVSKÝ, Samuel. *Context-aware filtr notifikací pro Android.* Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Szentandrási István.

Context-aware filtr notifikací pro Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Istvána Szentandrásiho. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Samuel Jaklovský
13.5.2016

Poděkování

Chtěl bych poděkovat svému vedoucímu práce Ing. Istvánovi Szentandrásimu za odbornou pomoc a usměrnění při psaní této práce, za jeho čas, cenné rady a informace a v neposlední řade za ochotu. Zvláštní díky patří mé manželce, rodičům a mým nejbližším za jejich podporu a pomoc, díky které sem mohl tuto práci dokončit.

Obsah

1 Úvod

S rýchlym rozvojom mobilných zariadení prišla aj potreba rozvíjať aplikácie, ktoré užívateľovi prácu s týmito zariadeniami uľahčujú. Pri takýchto aplikáciách je kladený dôraz na kontextovo špecifické služby, pri ktorých je kontext zvyčajne chápaný ako kombinácia údajov týkajúcich sa polohy, času, užívateľského profilu, profilu zariadenia, stavu siete a spôsobu použitia. Aplikácie poskytujúce kontextovo špecifické služby pomáhajú zrýchľovať a uľahčovať prácu s mobilnými zariadeniami. Využitie takýchto aplikácií sa môže týkať aj prispôsobenia používateľského rozhrania na základe prijímaného kontextu.

Cieľom tejto práce je vytvorenie aplikácie pre mobilné zariadenia používajúce operačný systém Android, ktorá bude na základe získaného kontextu prepínať medzi užívateľom predvolenými profilmi. V daných profiloch si užívateľ bude môcť vybrať predvolené nastavenia spôsobu upozornenia na prichádzajúce hovory, SMS správy a notifikácie aplikácií. Taktiež si bude môcť určovať výnimky, ktoré pod tieto nastavenia spadať nebudú.

Vo výsledku bude môcť používateľ automatizovať to, čo musí inak vykonávať mechanicky, a predísť vyrušovaniu neželanými telefonátmi a upozorneniami. Taktiež bude braný nemalý ohľad na to, aby bolo používateľské prostredie jednoduché a zrozumiteľné, čím uľahčí užívateľovi prácu so samotnou aplikáciou.

Tento dokument pojednáva o zhromaždených informáciách a dosiahnutých výsledkoch. Kapitola 2 vysvetľuje pojmy kontext, kontext-awareness a približuje podstatu kontext-aware systému. V kapitole 3 je popísané, akým spôsobom je zo získaného kontextu možné pomocou modelov rozhodovania vyhodnotiť aktuálny stav a situáciu. Kapitola 4 pojednáva o možnostiach získavania a následného použitia kontextu v operačnom systéme Android. Ďalej je táto kapitola venovaná návrhu používateľského rozhrania. Kapitola 5 ukazuje a analyzuje výsledky testovania. Stručný záver a plány pre budúcu prácu sú obsiahnuté v kapitole 6.

2 Kontext-awareness

Pojem kontext bol spojený s výpočtovou technológiou už po dlhú dobu. Od polovice 90. rokov bola stále väčšia pozornosť venovaná úlohe, ktorú môže kontext mať v prípade adaptívnych výpočtových technológií. Konkrétne bol záujem zameraný na to, ako je možné prispôsobiť počítačovú aplikáciu tak, aby zodpovedala požiadavkám alebo potrebám rôznych situácií a užívateľov. S narastajúcou prevahou výpočtových technológií orientovaných na služby sa schopnosť prispôsobiť sa stala synonymom pre prispôsobenie ponúkaných služieb. Očakáva sa teda, že kontext-aware systém dokáže prispôsobiť každú službu aktuálnemu kontextu použitia.[4]

2.1 Definícia kontextu

Slovo kontext nemá jednotnú definíciu. V Merriam-Webster's Collegiate Dictionary [1] je kontext zadefinovaný ako „*the interrelated conditions in which something exists or occurs*“, čiže ako nejaké prepojené podmienky, za ktorých niečo existuje alebo nastane. Na anglickej Wikipédii [2] je definovaný ako „*the surroundings, circumstances, environment, background, or settings that determine, specify, or clarify the meaning of an event or other occurrence*”, čiže okolie, okolnosti, životné prostredie, pozadie, alebo nastavenia, ktoré určujú, spresňujú, alebo objasňujú význam udalosti alebo iného výskytu. Tieto definície ale pozerajú na kontext zo všeobecného hľadiska.

Pokiaľ chceme definíciu kontextu z oblasti výpočtových technológií, tak Paul Prekop a Mark Burnett definujú kontext ako hocijakú informáciu, ktorá môže byť použitá na určenie situácie nejakej entity. Entita je definovaná ako človek, miesto, alebo objekt, ktorý je považovaný za relevantný v interakcii medzi užívateľom a aplikáciou, vrátane samotného užívateľa a aplikácie [3].

Existuje množstvo rozličných definícií a rozdelení kontextu. Okrem iného sa člení medzi fyzické a logické súradnice. To prvé predstavuje užívateľovu polohu a orientáciu, zatiaľ čo to druhé predstavuje aktuálnu úroveň detailov výslovne vyžiadanú užívateľom. Táto kategorizácia je tiež označovaná ako prevádzkový vs. koncepčný kontext. Koncepčný je užívateľ-centrický, zatiaľ čo prevádzkový je prostredie-centrický. Ďalší uhol pohľadu uvažuje o „vycítenom“ („*sensed*“), statickom, profilovanom alebo odvodenom kontexte, zatiaľ čo iný zastáva rozdelenie na kontext založený na stave a kontext založený na udalosti. Vo všetkých definíciách sa všeobecne kontext odkazuje na celú situáciu, ktorá je relevantná pre aplikáciu a jej užívateľov.

Pre túto prácu bolo zvolené, že kontext bude pozostávať z polohy, hluku, z toho, či zariadenie leží, či užívateľ chodí, z rýchlosti pohybu, z času a dňa v týždni.

2.2 Definícia kontext-awareness

Kontext-awareness je schopnosť programu alebo výpočtového zariadenia detekovať, vnímať, interpretovať, konať a reagovať na aspekty prostredia ako poloha, čas, teplota a užívateľská identita [5]. Alebo z hľadiska prispôsobenia aplikácie na kontext je to schopnosť aplikácií skúmať výpočtové prostredie a reagovať na dynamické zmeny ako poloha užívateľa, zbierka ľudí a dostupných zariadení v blízkosti, a prispôsobovať svoje správanie založené na kontexte aplikácie a prostredia [6].

Dey [7] porovnáva a sumarizuje vo svojej dizertačnej práci definície, ktoré boli dovtedy zaznamenané a poskytuje všeobecnejšiu a ľahšiu definíciu: „*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*“ Čiže systém je kontext-aware, ak používa kontext k poskytovaniu relevantných informácií a/alebo služieb pre používateľov, kde relevantnosť závisí na úlohe používateľa.

Kontext-awareness má za cieľ poskytnúť výpočtové zariadenia, ktoré získavajú kontextové informácie pomocou nových metód interakcie človeka s počítačom, alebo pomocou širokej škály dostupných snímačov. Výpočtové zariadenia vykonávajú vhodnú reakciu na správnom mieste a v správny čas prostredníctvom vyvodzovania zámeru užívateľa. V priebehu tohto procesu, a to aj v prípade, že užívateľ nezadá žiadnu informáciu, môže dostať správnu a rozumnú spätnú väzbu [8].

2.3 Kategorizácia kontext-awareness

Podľa vplyvu na správanie systému možno kontext-awareness rozdeliť na aktívny a pasívny.

- Aktívny – aplikácia sa automaticky prispôsobí získanému kontextu pomocou zmien v správaní aplikácie
- Pasívny – aplikácia predstavuje nový alebo aktualizovaný kontext používateľovi alebo kontext uchováva pre užívateľa, ktorý ho môže získať neskôr

2.4 Kontext-awareness systém

Kontext-awareness systém sa skladá z troch hlavných častí. V prvom rade systém zhromažďuje informácie o kontexte dostupné z používateľského rozhrania, vopred špecifikovaných údajov alebo snímačov a ukladá ich do úložiska. Okrem toho systém prevádza získané „surové“ kontextové informácie do zmysluplného kontextu, ktorý môže byť použitý. Nakoniec systém použije informácie, poskytne reakciu a odhalí príslušný kontext používateľovi.

Získavanie a snímanie kontextu

Získavanie kontextu je najzákladnejšia úroveň kontext-awareness. Vo všeobecnosti platí, že existujú tri spôsoby, ako získať kontext [9]:

- Nasnímaný kontext – informácie o prostredí a fyzikálne informácie možno získať cez fyzikálne alebo softvérové senzory. Informácie o návykoch užívateľa a záznamy o histórii sa získavajú väčšinou logickým senzorom hostiteľa.
- Odvodený kontext – tento druh kontextových informácií môže byť vypočítavaný za behu.
- Výslovne získaný kontext – napríklad preferencie používateľa možno získať pri výslovnej komunikácii so žiadajúcou aplikáciou

Napriek tomu, že je v systéme mnoho kontextových informácií, venujeme viac pozornosti identite, polohe, aktivite a času ako ostatným informáciám. Existujú viaceré spôsoby prístupu k informáciám o polohe. Najrozšírenejšia metóda je Global Positioning System (GPS). Ďalej sa dajú použiť infračervené a rádiové signály, RFID, WiFi, Bluetooth a vnútorný generátor GPS signálu. Taktiež je možné prijímať informácie o polohe zo siete určitej oblasti. Niektoré informácie o identite, ako napr. pohlavie užívateľa, sú do kontext-awareness vložené explicitne. Informácie o čase sa dajú jednoducho získať zo vstavaných hodín počítača. Zatiaľ čo kontextové informácie vysokej úrovni, ako napríklad aktuálna činnosť užívateľa, sú veľkou výzvou. Jedna možnosť pre ich získanie je strojové videnie, ktoré je založené na technológii fotoaparátu a spracovaní obrazu. Iným možným prístupom je nahliadnuť do kalendára používateľa a priamo zistiť, čo má užívateľ v danom čase robiť. Treťou metódou je použitie techník umelej inteligencie na rozpoznanie kompletného kontextu kombináciou niekoľkých jednoduchých nízkoúrovňových snímačov.

Modelovanie a reprezentácia kontextu

Na vylíčenie užívateľovho prostredia a situácie využíva kontext-awareness systém modelovania a reprezentácie kontextu. Modelovanie kontextu musí byť formulované tak, aby poskytovalo prostriedky, ktoré je aplikácia schopná použiť. Nie je problém zvládnuť všetky druhy kontextu modelovať samostatne, v dnešných technologických podmienkach je však veľká výzva zvládnuť všetky druhy kontextu v jednom všeobecnom kontextovom modelovaní. Podľa súčasnej situácie možno toto jednotné modelovanie kontextu rozdeliť do dvoch úrovní: Rôzne kontexty používajú rovnakú štruktúru dát na vyjadrenie režimu ako napr. Key-Value Models, Markup Scheme Models atď. Taktiež môže podporovať sémantickú jednotnosť, ako napr. Ontology Based Models, Graphical Models atď. [8]

Filtrovanie a zlučovanie kontextu

Surové zosnímané dáta sú nezjednotené, nestabilné a nepresné. Preto je potrebné efektívne filtrovať kontext a spájať na základe podmienok. Filtrovanie má za cieľ extrahovať významové informácie z dát snímača a zároveň odfiltrovať tie, ktoré systém nepotrebuje. Ďalším cieľom je filtrovať špecifické kontextové informácie z dôvodu zachovania súkromia.

Systém môže získať kontextové informácie rôznymi spôsobmi a môže sa stať, že sa informácie z rôznych zdrojov líšia, dokonca sú si niekedy protichodné. Z tohto dôvodu je potrebné na základe určitých pravidiel zlúčiť kontext tak, aby sme z nekonzistentných informácií získali v aplikácii maximálny prínos, a aby sa zabránilo nesprávnemu rozhodnutiu systému.

Ukladanie a znovuzískavanie kontextu

Surové kontextové dáta, tak ako aj filtrované a zlúčené kontextové informácie, možno ukladať pre neskoršie obnovenie. Kontext by mal byť usporiadaný do rôznych dátových štruktúr ako je tabuľka, objekt, strom, graf atď. Architektúra úložiska kontextu môže byť buď centralizovaná alebo distribuovaná.

Aplikácia kontextu

Dôležitou otázkou je ako efektívne využívať kontext. Dey [7] sumarizuje kategórie použitia kontextu a navrhuje tri, ktoré môžu byť kontextovými aplikáciami podporované:

- Prezentácia informácií a služieb užívateľom
- Automatické spustenie služby
- Značenie kontextu k informáciám pre neskoršie obnovenie a použitie

3 Model rozhodovania

Dáta získané zo senzorov je potrebné, po ich správnom odfiltrovaní, analyzovať a dedukovať z nich potrebný kontext. Na túto dedukciu sa používajú modely rozhodovania.

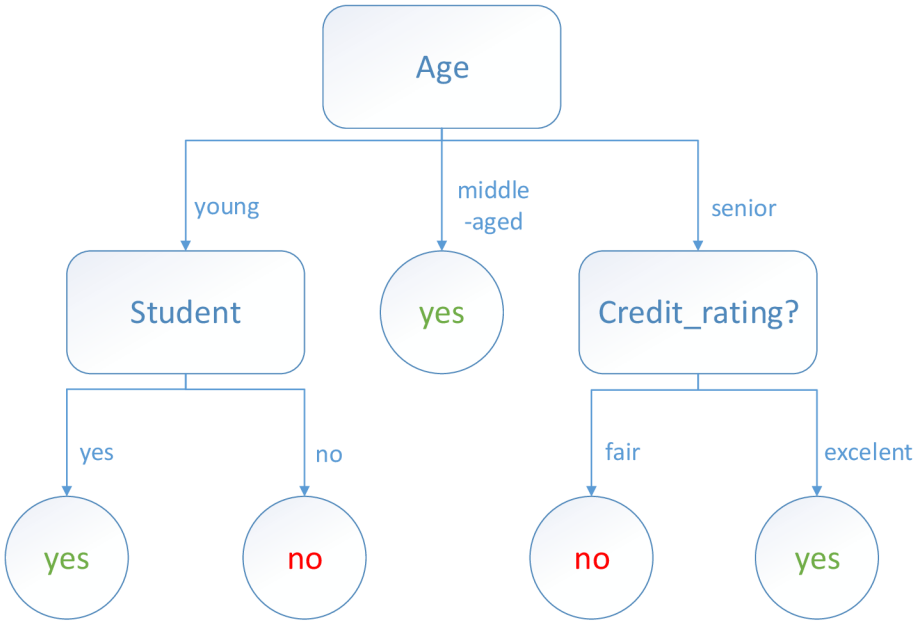
Existuje veľké množstvo modelov rozhodovania, ktoré sa v kontext-awareness systémoch používajú. Použitie konkrétnych z nich alebo dokonca kombinácií týchto modelov závisí od ich zložitosti, typu získavaného kontextu, pôvodu informácií, ale aj užívateľských preferencií.

Vo všeobecnosti pozostáva proces vytvárania rozhodovacích modelov z dvoch hlavných častí. Prvým krokom je učenie alebo tréning klasifikátoru. V tomto procese je vytvorená tréningová množina klasifikácii, teda príkladov dát získaných zo senzorov, priradených ku konkrétnym triedam. Z tejto tréningovej množiny potom klasifikátor vychádza pri druhom kroku, ktorým je klasifikácia. Na základe naučených klasifikácii spracováva nové prichádzajúce dáta a určuje, s akou presnosťou patria do tej-ktorej triedy.

V nasledujúcich podkapitolách sa nachádza popis rozhodovacích modelov, ktoré boli použité pri implementácii navrhnutého kontext-awareness systému.

3.1 Rozhodovací strom

Rozhodovací strom je klasifikátor so stromovou štruktúrou. Uzly, ktoré nie sú listami, sa nazývajú rozhodovacie. Tieto uzly predstavujú pridelený atribút a vetvy týchto uzlov sa zhodujú s možnými hodnotami tohto atribútu. Počet vetiev je konečný, a preto musia atribúty nadobúdať len diskrétne hodnoty. Spojité hodnoty je preto potrebné diskretizovať na konečný počet intervalov. Listy stromu predstavujú triedy, do ktorých má byť súbor atribútov klasifikovaný.



Obrázok 1: Príklad jednoduchého rozhodovacieho stromu použitého pre kontext "kúpi počítač" [14]

Klasifikácia začína v koreni stromu, ktorý je taktiež rozhodovacím uzlom. V rozhodovacom uzle sa zistí hodnota atribútu a podľa nej sa pokračuje po príslušnej vetve do ďalšieho uzlu. Tento proces sa opakuje, až kým nie je na konci vetvy list. Na základe toho, ku ktorému listu sa rozhodovací strom dopracoval, pridelí ku klasifikovanému súboru atribútov triedu.

3.2 Naïve Bayes

Naïve Bayes je jedna z najrozšírenejších klasifikačných techník. Na základne Bayesovej teorémy predpovedá pravdepodobnosť, s ktorou daný prípad patrí do konkrétnej triedy. Naïve Bayes klasifikátor je založený na predpoklade, že jednotlivé atribúty sú od seba nezávislé. Čiže efekt, ktorý má hodnota ktoréhokoľvek atribútu na danú triedu, nie je ovplyvnený hodnotami ostatných atribútov. A z tohto pohľadu je táto klasifikačná technika považovaná za „naivnú“.[10]

Napríklad ovocie môže byť považované za pomaranč, pokiaľ je oranžové, guľaté a v priemere má približne 15 centimetrov. Aj keď tieto vlastnosti od seba závisia, Naïve Bayes klasifikátor uvažuje, že všetky tieto vlastnosti sa nezávisle podieľajú na pravdepodobnosti, že dané ovocie je pomaranč.

Pokiaľ to aplikujeme na kontext-awareness, tak existuje *m* odlišných tried, ktoré táto aplikácia rozoznáva, napríklad: domov, práca, v meste, atď. Na klasifikáciu máme vstupný vektor a tento vektor je potrebné priradiť ku konkrétnej triede. Tento vstupný vektor predstavuje jednotlivé získané atribúty. V kontext-aware programovaní väčšinou tento vektor pozostáva z hodnôt získaných pomocou senzorov a prijímačov. Naïve Bayes určí zvlášť pre každý z týchto atribútov pravdepodobnosť, že patria do niektorej z tried, a aplikuje Bayesovu teorému na zistenie, ktorá trieda získala najvyššiu pravdepodobnosť.[11]

Nakoľko je tento model rozhodovania v implementácii použitý ako hlavný rozhodovací model pre dedukciu kontextu, jeho činnosť bude popísaná detailnejšie. (Adaptované z [11], [12], [13])

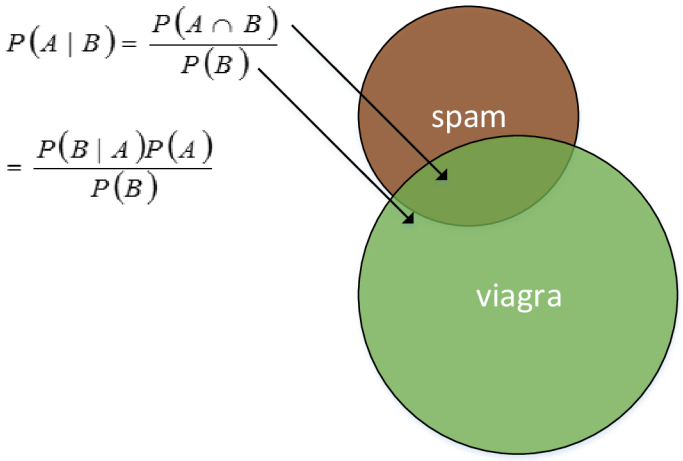
3.2.1 Bayesova teoréma

Pre vysvetlenie Bayesovej teorémy je potrebné zadefinovať tieto pojmy:

- *Príklad (**X**)* je nový, nezaradený vstup. V našom prípade sa jedná o sadu hodnôt získanú meraním, jednu pre každý atribút kontextu (poloha, hluk, deň, hodina ...)
- *Hypotéza (**H**)*, že nejaký príklad *X* patrí do konkrétnej triedy.
- *Trieda (**C**)* je označenie priradené k sade hodnôt atribútov. Triedou je napríklad domov, práca alebo podnik.
- *Vierohodnosť (**P(X|H)**)* je pravdepodobnosť príkladu *X* v prípade, že platí hypotéza *H*. Napríklad pravdepodobnosť, že bude nameraná konkrétna sada hodnôt v prípade, že platí, že sa užívateľ nachádza doma.
- *Apriórna pravdepodobnosť (**P(H)**)* je pravdepodobnosť hypotézy *H* bez uvažovania príkladu *X*. Čiže napríklad pravdepodobnosť toho, že užívateľ je doma, bez toho, aby bolo prihliadnuté na polohu, hluk, deň a ostatné atribúty kontextu.
- *Posteriórna pravdepodobnosť (**P(H|X)**)* je pravdepodobnosť, že platí hypotéza *H*, pokiaľ platí príklad *X*. Čiže napríklad pravdepodobnosť toho, že sa užívateľ nachádza doma, pokiaľ je nameraná konkrétna sada hodnôt. Na základe Bayesovej teorémy je možné posteriórnu pravdepodobnosť $P(H|X)$ dopočítať podľa nasledujúceho vzorca:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

- Pričom **P(X)** je *normalizačná konštanta* zabezpečujúca, aby sme vo výsledku dostali pravdepodobnosť.

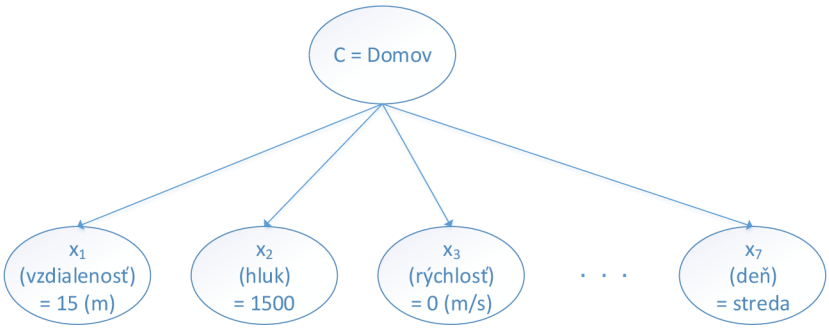


Obrázok 2: Posteriórna pravdepodobnosť, že e-mail bude spam, pokiaľ sa v ňom nachádza slovo viagra [15]

3.2.2 Naïve Bayes klasifikátor

Algoritmus Naïve Bayes klasifikátora pracuje potom nasledovne:

1. Nech D je tréningová množina príkladov X zaradených do tried C. Každý príklad X je reprezentovaný n-ticou hodnôt $x_1, x_2, ..., x_n$ patriacich n atribútom $A_1, A_2, ..., A_n$. Majme m tried $C_1, C_2, ..., C_m$ definujúcich rôzne kontextové situácie.



Obrázok 3: Príklad X z tréningovej množiny D

2. Príklad X bude priradený triede C_i s najväčšou posteriórnou pravdepodobnosťou:

$$X \in C_i \leftrightarrow P(C_i|X) > P(C_j|X)\ pre\ 1 \leq j \leq m, j \neq i$$

3. Vychádzame z Bayesovej teorémy:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Keďže $P(X)$ je konštantná pre všetky triedy C_i, potrebujeme získať maximum pre $P(X|C_i)P(C_i)$. $P(C_i)$ získame pomocou vzťahu:

$$P(C_i) = \frac{s_i}{S},$$

kde s_i predstavuje počet príkladov z tréningovej množiny D patriacich do triedy C_i a S je počet všetkých príkladov v tréningovej množine D. Pravdepodobnosť $P(X|C_i)$ dokážeme určiť z tréningovej množiny D ako:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i)\ \text{x}\ P(x_2|C_i)\ \text{x}\ \ldots\ \text{x}\ P(x_n|C_i)$$

- Pokiaľ atribút A_k nadobúda diskrétne hodnoty, potom

$$P(x_k|C_i) = \frac{s_{ik}}{s_i}.$$

Pričom s_{ik} je počet príkladov z tréningovej množiny patriacich triede C_i, pre ktoré $A_k = x_k$.

- Pokiaľ atribút A_k nadobúda spojité hodnoty, používa sa väčšinou Gaussovo normálne rozloženie:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma_{C_i}} e^{-\frac{(x_k-\mu_{C_i})^2}{2\sigma_{C_i}^2}}$$

Kde μ_{C_i} je stredná hodnota a σ_{C_i} je rozptyl hodnôt atribútu A_k z príkladov patriacich triede C_i.

4. Príklad X je teda zaradený do triedy, pre ktorú je pravdepodobnosť $P(X|C_i)P(C_i)$ najväčšia.

3.2.3 Laplaceova korekcia

Pri použití tohto algoritmu môže dôjsť k jednému častému problému. A to v prípade, že pravdepodobnosť $P(X|C_i)$ je rovná 0. Tento problém nastáva, keď pre danú triedu neexistuje v množine dát žiaden príklad.

Riešením je Laplaceova korekcia, čiže pridanie jedného prvku do každej množiny. Vďaka tomu atribút, ktorý sa v tréningovej množine nevyskytuje, dostane pravdepodobnosť, ktorá je nenulová, ale veľmi nízka.

4 Návrh implementácie

Ako už bolo spomenuté, cieľom tejto práce je vytvorenie aplikácie pre mobilné zariadenia používajúce operačný systém Android, ktorá bude na základe získaného kontextu prepínať medzi užívateľom predvolenými profilmi. Pod kontextom sa rozumejú informácie zo senzorov, ktoré poskytujú predstavu o prostredí, v akom sa užívateľ nachádza, a taktiež informácie ako čas a deň v týždni, ktoré poskytujú širší obraz o možných užívateľových aktivitách.

V daných profiloch si užívateľ bude môcť vybrať predvolené nastavenia spôsobu upozornenia na prichádzajúce hovory, SMS správy a notifikácie aplikácií. Bude si môcť určiť, či má telefón zotrvať v tichom režime, v režime vibrovania alebo v hlasnom režime. Akonáhle systém vyhodnotí, že sa užívateľ nachádza v konkrétnom profile, upravia sa nastavenia zvuku tak, ako si to užívateľ predtým navolil.

Taktiež si bude môcť určovať výnimky, ktoré pod tieto nastavenia spadať nebudú. Čiže ak bude napríklad potrebovať, aby bol na prichádzajúci hovor od konkrétnej osoby upozornený, aj keď sa nachádza v profile, v ktorom má nastavený tichý režim, určí si pre túto osobu výnimku s nastavením zvuku na hlasný režim.

Ako posledná skupina nastavení budú časové predvoľby. V nich si bude môcť používateľ určiť, v ktorých časových úsekoch sa aplikácia bude riadiť inak ako na základe profilov. V úseku zadefinovanom časovou predvoľbou sa bude prihliadať len na nastavenie priradené k tejto predvoľbe. Systém sa nebude zaujímať o to, v akom profile sa užívateľ aktuálne nachádza.

Pre dosiahnutie funkčnosti tohto projektu som musel vykonať nasledovné:

- Zistiť, aké možnosti ponúka operačný systém Android pre získavanie údajov na zostavenie kontextu, a následne navrhnúť a implementovať časť aplikácie, ktorá tieto údaje bude na pozadí získavať.
- Navrhnúť a implementovať časť aplikácie (klasifikátor), ktorá bude tieto údaje spracovávať a vyhodnocovať pravdepodobnosť pre jednotlivé profily.
- Navrhnúť a implementovať prehľadné, jednoduché a prívetivé užívateľské rozhranie.

Ako vývojové prostredie bolo zvolené prostredie Android Studio [22]. Toto prostredie bolo zvolené na základe jeho kompatibility, jednoduchosti používania a možností a funkcií, ktoré zjednodušujú a sprehľadňujú prácu pri vývoji aplikácie pre operačný systém Android.

Za názov aplikácie som si zvolil NotiMan. Tento názov predstavuje zlúčenie skratiek slovného spojenia v angličtine: *Notification Manager*, čo v preklade znamená správca upozornení.

Informácie o jednotlivých možnostiach, funkciách, riešeniach, triedach a rozšíreniach som čerpal z oficiálnych stránok pre Android a Google developerov (Android Developer [17], Google Developers [16]) a taktiež z fór, na ktorých iní vývojári predkladajú svoje problémy a riešenia z oblasti vývoja.

4.1 Získavanie a snímanie kontextu

Ako už bolo spomenuté vyššie, v podkapitole 2.1, kontext, na základe ktorého bude táto aplikácia pracovať, bude pozostávať z polohy užívateľa, konkrétne zo vzdialenosti k záujmovým bodom (domov, práca), z hladiny hluku, z toho, či je zariadenie položené, či užívateľ chodí, z rýchlosti pohybu, z času a dňa v týždni.

Poloha užívateľa

Pre získavanie polohy užívateľa som použil možnosti ponúkané od Google Play Services, konkrétne LocationServices, ktoré poskytuje GoogleApiClient. LocationServices poskytujú možnosť získať aktuálnu polohu zariadenia, ktorá je vyhodnocovaná na základe GPS a mobilných sietí. V pravidelnom intervale budem túto aktuálnu polohu pomocou triedy LocationRequest získavať a ukladať. Interval zisťovania polohy bude nastavený na 1 minútu. Ak by bol interval príliš krátky, malo by to za následok príliš vysokú spotrebu batérie. Aby bola získavaná poloha s najvyššou možnou presnosťou, nastavím prioritu na PRIORITY_HIGH_ACCURACY.

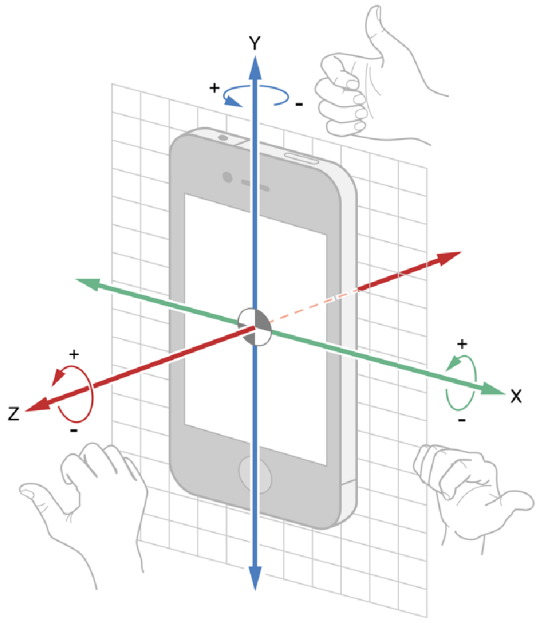
Hladina hluku

Na zistenie hladiny hluku z okolia bude použitý mikrofón zariadenia. Pre prístup k tomuto senzoru využijem triedu MediaRecorder. Táto trieda umožňuje nahrávať audio a video z mikrofónu a kamery zariadenia. V tomto prípade bude využité iba nahrávanie audia z mikrofónu zariadenia, zachytená nahrávka ale nebude nikde ukladaná, nakoľko pre potreby aplikácie nie je potrebná. Potrebné údaje získam pomocou funkcie getMaxAmplitude(). Táto metóda vracia maximálnu amplitúdu hladiny hluku nameranú od chvíle, kedy bola naposledy zavolaná.

Orientácia zariadenia a chôdza užívateľa

Pre získanie údajov o tom, či je telefón položený a či užívateľ chodí, využijem triedy SensorManager a SensorListener. Konkrétne použijem údaje zo senzorov Sensor.TYPE_ACCELEROMETER (pre výpočet orientácie zariadenia) a Sensor.TYPE_STEP_DETECTOR (na zistenie, či užívateľ chodí).

Senzor typu accelerometer poskytuje údaje o sile pôsobiacej na jednotlivé osi zariadenia (viď. Obrázok 4). Na základe veľkosti týchto síl (je potrebné vziať v úvahu aj gravitačné zrýchlenie) je možné dopočítať orientáciu zariadenia, a teda či je v polohe ležmo alebo je nejakým spôsobom natočené inak.



Obrázok 4: Osi orientácie a pohybu mobilného zariadenia [18]

Senzor typu step_detector poskytuje údaje o tom, či bol detekovaný krok. Pri každom kroku je na tento krok upozornené. Pokiaľ tento senzor rozpozná krok, je považované, že užívateľ chodí, pokiaľ prejde 10 sekúnd bez toho, aby bol rozpoznaný krok, je považované, že užívateľ práve nevykonáva činnosť chôdze.

Rýchlosť užívateľa

Rýchlosť užívateľa bude získavaná tak isto, ako aj jeho poloha pomocou LocationServices. V polohe, ktorú poskytuje, je zakomponovaná aj rýchlosť, vypočítaná na základe vzdialenosti od poslednej známej polohy a času, ktorý od zistenia tejto polohy uplynul.

Čas a deň v týždni

Na získavanie času a dňa v týždni použijem inštancie triedy Calendar (pre získanie dňa) a Date (pre získanie času), ktoré pracujú so systémovým časom zariadenia.

4.1.1 Organizácia snímania kontextu

Všetko vyššie spomenuté získavanie údajov bude prebiehať na pozadí aplikácie a bude ho riadiť inštancia triedy BackgroundService, ktorá rozširuje triedu Service, slúžiacu na takéto účely (riadenie na pozadí). V BackgroundService budú získané údaje ukladané a v intervaloch posielané na vyhodnotenie do triedy NaiveBayesClassifier. Tá predstavuje Naive Bayes klasifikátor, ktorý vyhodnocuje získaný kontext pre túto aplikáciu a určuje pravdepodobnosť, že sa užívateľ nachádza v konkrétnom profile. Pravidelné odosielanie na vyhodnotenie bude zabezpečovať trieda AlarmManager.

4.1.2 Výnimky

V každom profile bude mať užívateľ možnosť nastaviť si výnimky, pre ktoré zvukové nastavenia daného profilu nebudú platiť. Užívateľ si jednoducho vyberie niekoho spomedzi kontaktov a pridelí mu iné zvukové nastavenie. V prípade prichádzajúceho hovoru sa porovná, či hovor prichádza od niekoho zo zoznamu výnimiek pre aktuálny profil. Pokiaľ bude nájdená zhoda, tak sa na prichádzajúci hovor aplikujú zvukové nastavenia z danej výnimky.

4.1.3 Časové predvoľby

Užívateľ bude mať taktiež možnosť nastaviť si časové predvoľby. Tento prvok spočíva v tom, že si užívateľ nastaví istý časový úsek a jeho opakovanie v rámci dní v týždni. Pre takúto časovú predvoľbu si bude môcť následne zvoliť osobitné nastavenie zvuku, ktoré bude aplikované v rámci daného časového úseku.

Časové predvoľby budú nadradené profilom. To znamená, že pokiaľ bude zistené, že má byť aplikovaná časová predvoľba, systém aplikuje zvukové nastavenie tejto predvoľby bez ohľadu na získaný kontext a príslušnosť k akémukoľvek profilu. Z tohto dôvodu nebudú, pre zníženie réžie, pri zistení platnosti časovej predvoľby posielané aktuálne namerané hodnoty na vyhodnocovanie do klasifikátora. Zisťovanie, či má byť aplikovaná časová predvoľba, bude prebiehať pomocou rozhodovacieho stromu, ktorý porovná aktuálny čas a deň v týždni s existujúcimi predvoľbami a v prípade zhody informuje systém o nastavení zvuku, ktoré má byť aplikované.

4.2 Vyhodnocovanie kontextu

Keď už je kontext nasnímaný, možno prejsť k jeho vyhodnoteniu. Vyhodnocovanie kontextu bude mať na starosti trieda NaiveBayesClassifier. Táto trieda bude pracovať na princípe Naive Bayes klasifikátora a na základe uloženej tréningovej sady bude vyhodnocovať dáta zo vstupu a určovať pravdepodobnosť, že tieto dáta spadajú pod konkrétnu triedu (profil). Konkrétne bude klasifikátor rozhodovať medzi týmito piatimi profilmi: Domov, Práca, V meste, Podnik a Cestovanie.

Tréningová sada bude pozostávať z databázy naplnenej príkladmi, ktoré budú reprezentované inštanciami triedy Classification. Každá inštancia tejto triedy predstavuje jednu klasifikáciu. Obsahuje teda konkrétne dáta a triedu (profil) priradenú k týmto dátam. Rozsah počiatočnej tréningovej sady som zvolil v počte príkladov 500. Pre každý profil 100 príkladov. Menej príkladov by nezabezpečovalo optimálne pokrytie variácie dát pre každý profil a väčší počet príkladov by naopak sťažil možnosť užívateľa efektívne „učiť“ klasifikátor. Ak by bola totiž tréningová sada priveľmi rozsiahla, užívateľom uložené príklady by mali príliš nízky efekt na výslednú pravdepodobnosť.

Všetky klasifikácie budú uložené v objektovo orientovanej databáze Realm [19]. Túto databázu som zvolil na základe výhody možnosti ukladania objektov (inštancií) vlastných tried. Taktiež bola zvolená pre jej implementované metódy na správu a vyhľadávanie v rámci databázy, výrazne lepšiu prístupovú rýchlosť oproti SQLite [20] a tiež jednoduchosť implementácie a použitia.

Vstupom pre klasifikátor bude zemepisná šírka a dĺžka udávajúca aktuálnu polohu, hladina hluku, údaj o tom, či sa zariadenie nachádza v polohe ležmo, či boli zaznamenané kroky, čiže či užívateľ chodí, rýchlosť pohybu zariadenia, aktuálny časový interval a deň v týždni. Pre zjednodušenie a zovšeobecnenie práce klasifikátora budú všetky údaje reprezentované číselnými hodnotami a ukladané ako double. Pri orientácii zariadenia predstavuje hodnota 1.0 polohu ležmo, a 0.0 inú polohu. Pri údaji o tom, či užívateľ chodí znamená hodnota 1.0 pravdivosť tohto výroku a 0.0 jeho nepravdu. Pre aktuálny čas bolo zvolené rozdelenie do šiestich intervalov 1.0, 2.0, ..., 6.0, pričom každý interval predstavuje úsek štyroch hodín. Dňom v týždni sú taktiež pridelené numerické hodnoty, 1.0 pre nedeľu, 2.0 pre pondelok až 7.0 pre sobotu.

Ako prvý krok sa vypočíta vzdialenosť v metroch medzi aktuálnou polohou a záujmovými bodmi (adresa pre domov a prácu). Následne sa odošlú všetky údaje na výpočet pravdepodobnosti pre príslušnosť ku každej triede. V prípade tried rôznych od triedy domov a práca, bude k atribútu vzdialenosť rovno pridelená konštantná pravdepodobnosť. Vypočítané pravdepodobnosti sa zoradia a na výstup sa pošle názov triedy, pre ktorú majú údaje zo vstupu najvyššiu pravdepodobnosť.

4.3 Návrh užívateľského rozhrania

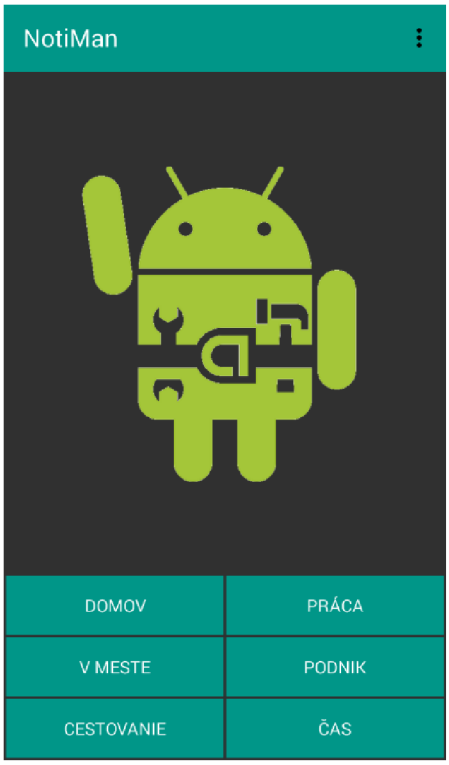
Pri užívateľskom rozhraní bude kladený dôraz na jednoduchosť, prehľadnosť a intuitívnosť. Je dôležité, aby sa užívateľ dokázal v užívateľskom rozhraní ľahko orientovať. Taktiež musí byť očividné a užívateľ musí sám pochopiť, ktoré okno a tlačidlo na čo slúži a ako sa celá aplikácia ovláda.

Na implementáciu užívateľského rozhrania budú z najväčšej časti použité triedy Activity a Fragment. V zjednodušenom ponímaní trieda Activity zabezpečuje prevažne beh na pozadí okna (napr. funkcie spustené po kliknutí na nejaké tlačidlo), Fragment má na starosti zobrazovanie jednotlivých vnútorných prvkov a interakciu s užívateľom. Zatiaľ čo každé okno bude predstavovať samostatná inštancia triedy rozširujúcej triedu Activity, nie každé okno musí použiť osobitný Fragment. V prípade okien, ktoré majú rovnaké usporiadanie vnútorných prvkov (textových polí, tlačidiel, atď.), je možné využiť inštancie rovnakej triedy, ktorá rozširuje triedu Fragment.

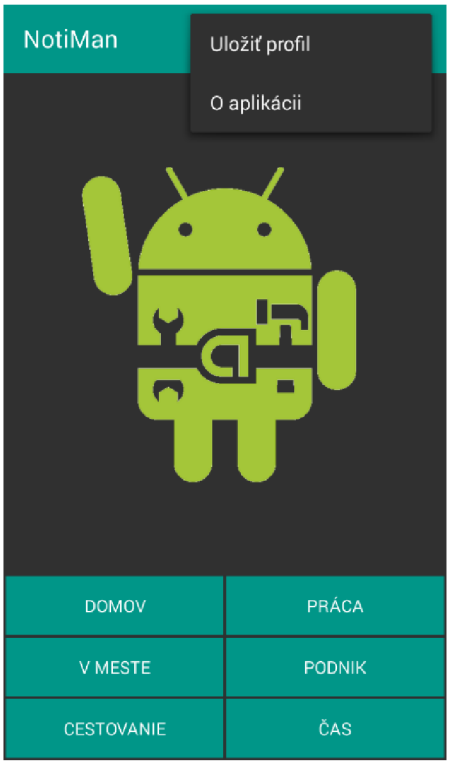
Užívateľské rozhranie bude pozostávať z týchto okien a prvkov:

Úvodná obrazovka

Úvodná obrazovka nebude poskytovať žiadne informácie, bude v podstate len poskytovať možnosť prejsť na konkrétny profil (Domov, Práca, V meste, Podnik a Cestovanie) alebo časové predvoľby. Tieto možnosti budú ponúknuté v podobe jednoduchých tlačidiel v spodnej časti obrazovky (viď. Obrázok 6). Taktiež bude možné pri rozkliknutí menu prejsť do informácií o aplikácii a k možnosti uložiť aktuálny profil (viď. Obrázok 5).



Obrázok 6: Úvodná obrazovka.

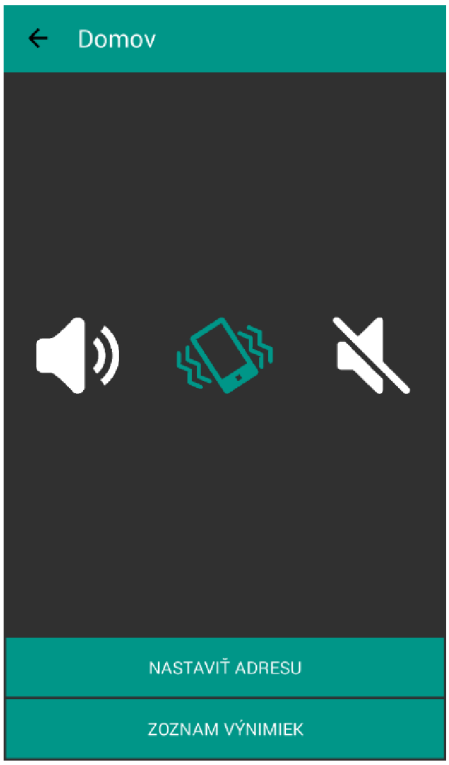


Obrázok 5: Rozkliknuté menu na úvodnej obrazovke.

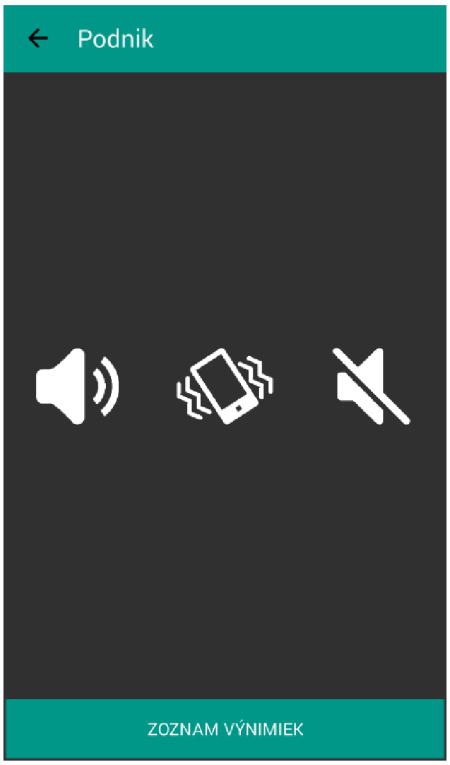
Hlavná obrazovka profilu

Po rozkliknutí konkrétneho profilu sa zobrazia možnosti nastavenia zvuku pre daný profil. Tieto možnosti budú zobrazené pomocou troch obrázkových tlačidiel zobrazujúcich symboly pre dané zvukové nastavenia, pričom farebne zvýraznený symbol predstavuje aktuálne zvolené nastavenie (viď. Obrázok 8). Nastavenie zvuku je možné prepínať jednoduchým kliknutím na iný symbol, prípadne zrušiť kliknutím na aktuálne zvolený symbol. Keď nie je zvolené žiadne nastavenie (viď. Obrázok 7), znamená to, že užívateľ si neželá, aby boli jeho zvukové nastavenia v danom profile upravované.

V spodnej časti obrazovky sa nachádza tlačidlo, ktorým môže užívateľ prejsť do zoznamu výnimiek (viď. Obrázok 7). V prípade profilu Domov a Práca sa nad týmto tlačidlom nachádza ešte tlačidlo, ktoré umožňuje prejsť na nastavenie adresy (viď. Obrázok 8).



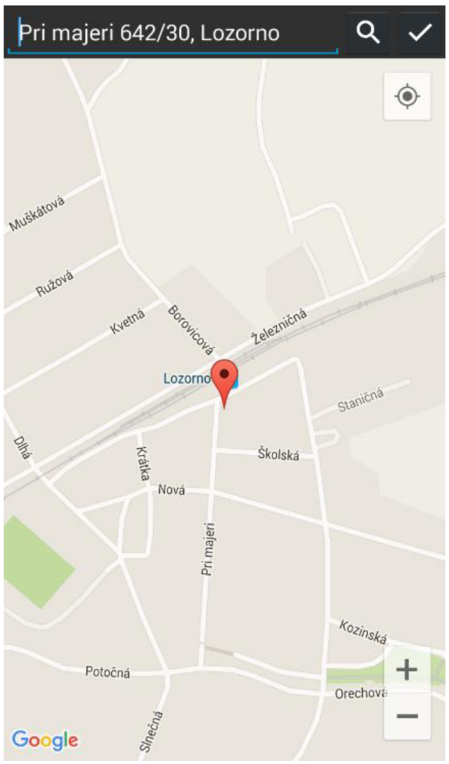
Obrázok 8: Hlavná obrazovka profilu Domov so zaškrtnutým nastavením zvuku na vibrovať.



Obrázok 7: Hlavná obrazovka profilu Podnik v prípade nezvolenia žiadneho zvukového nastavenia.

Obrazovka nastavenia adresy

V profiloch Domov a Práca sa po kliknutí na tlačidlo *Nastaviť adresu* zobrazí okno s mapou. V hornej časti obrazovky sa nachádza upravovateľné textové pole pre zadávanie adresy. Vedľa tohto poľa sú dve tlačidlá. Jedno pre vyhľadanie zadanej adresy a druhé pre potvrdenie danej adresy a uloženie do systému. V prípade, že je adresa nastavená, sa automaticky po otvorení okna daná adresa zobrazí v textovom poli a na mape. Pokiaľ chce užívateľ adresu odstrániť, vymaže adresu z textového poľa, nechá ho prázdne a stlačí tlačidlo potvrdenia adresy. V rámci mapy môže užívateľ taktiež dať vyhľadať svoju aktuálnu polohu. Priblížiť a oddialiť môže buď zaužívanými gestami alebo pomocou implementovaných tlačidiel. (viď. Obrázok 9)

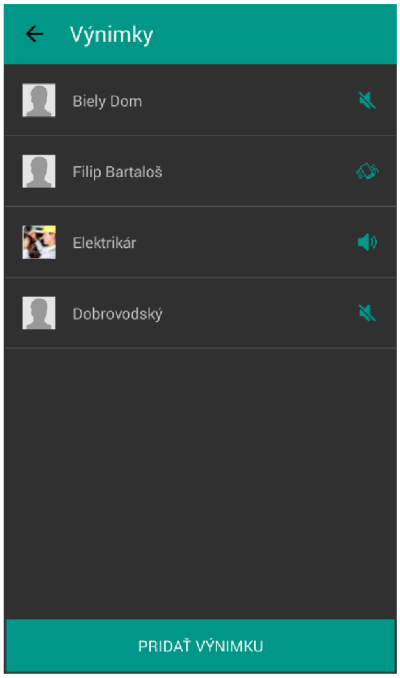


Obrázok 9: Obrazovka nastavenia adresy.

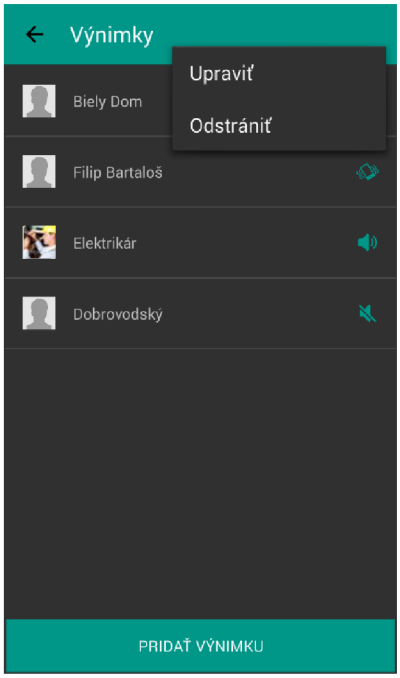
Obrazovka zoznamu výnimiek a pridávania výnimiek

Po kliknutí na tlačidlo *Zoznam výnimiek* v hlavnej obrazovke profilu sa zobrazia aktuálne nastavené výnimky pre daný profil. Výnimky sú zobrazované vo forme zoznamu, pričom každá položka zoznamu zobrazuje fotografiu kontaktu, meno kontaktu a symbol nastavenia zvuku, ktoré je k danému kontaktu pridelené (viď. Obrázok 12). Pre potreby zobrazenia tohto zoznamu bude použitá trieda rozloženia `ListView` a o naplnenie jednotlivých položiek konkrétnymi kontaktmi sa postará adaptér. Ako adaptér bude použitá vlastná trieda rozširujúca triedu `CursorAdapter`. Pri kliknutí na niektorý z kontaktov v zozname výnimiek sa roztvorí menu, v ktorom si užívateľ môže vybrať, či chce danú výnimku upraviť alebo odstrániť zo zoznamu (viď. Obrázok 11). V dolnej časti obrazovky sa nachádza tlačidlo *Pridať výnimku*, ktoré presmeruje užívateľa na zoznam kontaktov v jeho zariadení.

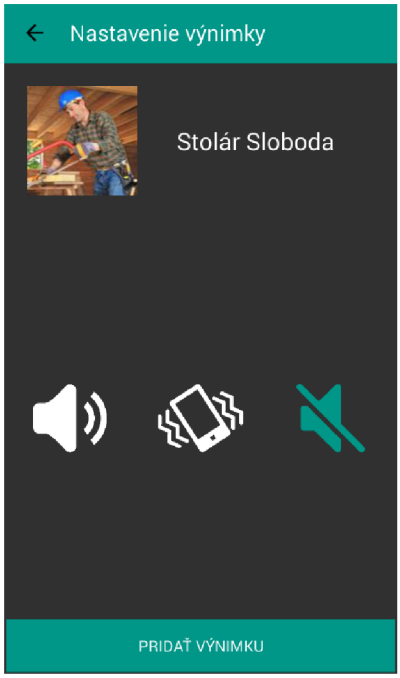
Po kliknutí na niektorý z kontaktov sa otvorí okno nastavenie výnimky. V tomto okne sú zobrazené údaje o kontakte, konkrétne fotografia a meno. Uprostred okna sú zobrazené symboly pre jednotlivé zvukové nastavenia. Užívateľ musí zvoliť jedno z týchto nastavení kliknutím na daný symbol. Zvolené nastavenie sa prejaví farebným zvýraznením symbolu. V spodnej časti obrazovky sa nachádza tlačidlo *Pridať výnimku*, ktorým užívateľ potvrdí uloženie danej výnimky do systému. (viď. Obrázok 10)



Obrázok 12: Zoznam výnimiek.



Obrázok 11: Menu zobrazené pri kliknutí na výnimku.

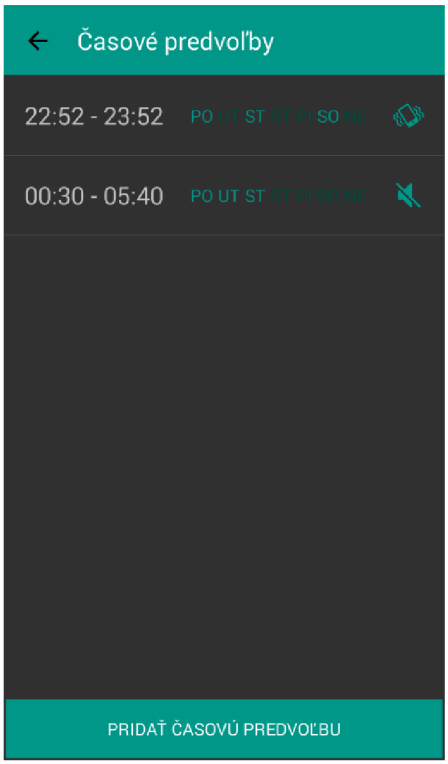


Obrázok 10: Nastavenie výnimky.

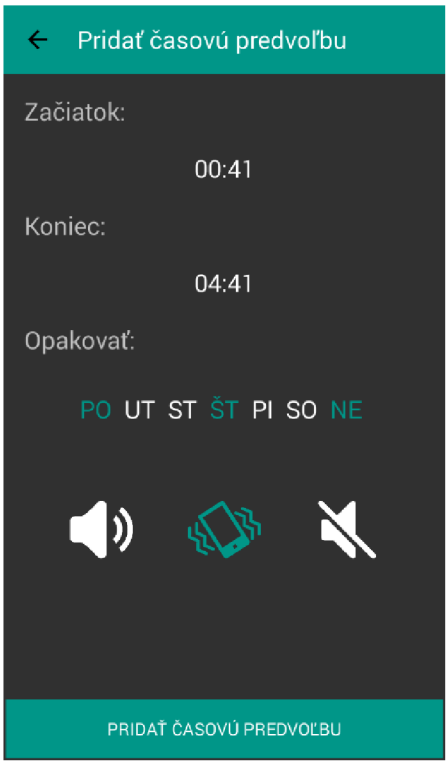
Obrazovka časových predvolieb a pridávania časových predvolieb

Po kliknutí na tlačidlo *Čas* na úvodnej obrazovke sa zobrazí zoznam časových predvolieb. Každá položka na zozname predstavuje jednu časovú predvoľbu. V rámci zobrazenia pozostáva predvoľba z počiatočného času, konečného času, dní v týždni (zvolené dni, počas ktorých sa má táto časová predvoľba opakovať, sú farebne odlíšené) a symbolu nastavenia zvuku (viď. Obrázok 14). Tak ako pri zozname výnimiek, aj v tomto prípade bude použitá trieda rozloženia ListView a adaptér rozširujúci triedu CursorAdapter. Pri kliknutí na niektorú predvoľbu sa taktiež zobrazí menu umožňujúce užívateľovi úpravu a odstránenie danej predvoľby. V dolnej časti obrazovky sa nachádza tlačidlo *Pridať časovú predvoľbu*, ktoré užívateľa presmeruje do nového okna, v ktorom si môže navoliť novú predvoľbu.

V okne pre pridávanie časovej predvoľby si užívateľ zvolí počiatočný a konečný čas, dni v týždni, kedy sa má táto predvoľba opakovať, a nastavenie zvuku. Nastavenie zvuku volí pomocou kliknutia na symbol pre konkrétne nastavenie. V spodnej časti obrazovky sa nachádza tlačidlo, ktorým túto predvoľbu potvrdí a uloží. (viď. Obrázok 13)



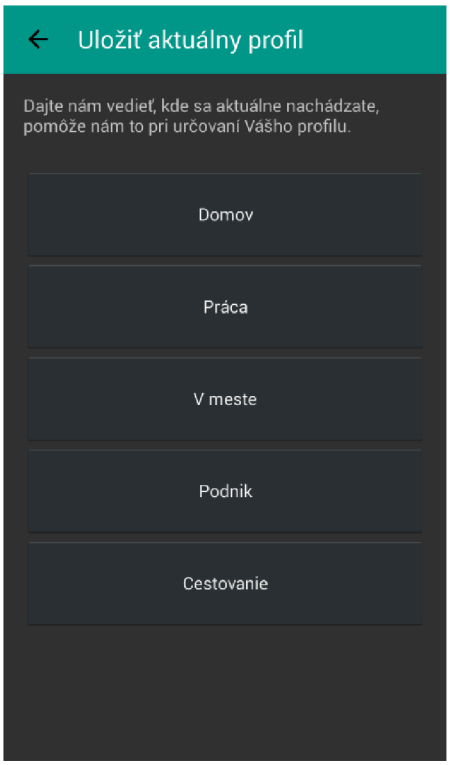
Obrázok 14: Obrazovka so zoznamom časových predvolieb.



Obrázok 13: Okno pre pridávanie časovej predvoľby.

Obrazovka pre ukladanie aktuálneho profilu

Po kliknutí na položku *Uložiť profil* v menu úvodnej obrazovky sa zobrazí okno, v ktorom môže užívateľ udať, kde sa práve nachádza, čiže aký by mal byť jeho aktuálny profil. Učiniť tak bude môcť kliknutím na tlačidlo s názvom daného profilu (viď. Obrázok 15). Po kliknutí sa do tréningovej sady v databáze pridá nová klasifikácia s aktuálnymi nameranými hodnotami a profilom, ku ktorému majú byť tieto hodnoty pridelené. Pridávaním nových klasifikácii, z ktorých klasifikátor vychádza pri výpočte pravdepodobnosti, je možné pre podobné namerané hodnoty upraviť výsledné pravdepodobnosti pre príslušnosť ku konkrétnej triede (profilu). Takýmto spôsobom je umožnené užívateľovi „učiť" klasifikátor, pokiaľ nepracuje úplne presne.

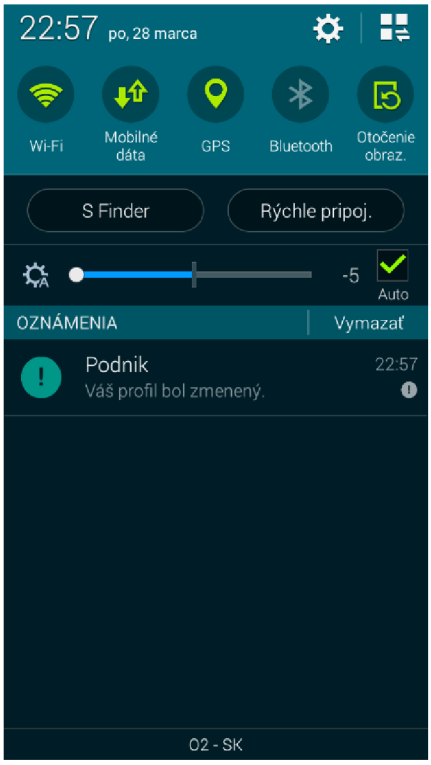


Obrázok 15: Obrazovka ukladania aktuálneho profilu.

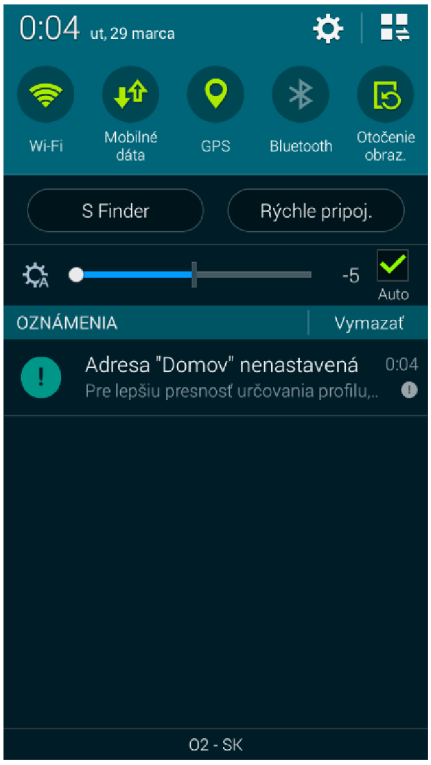
Notifikácie

O zmene aktuálneho profilu je užívateľ upovedomený pomocou notifikácií. Pre účel notifikácií bude použitá trieda NotificationManager. V notifikácii sa nachádza upozornenie na to, že profil bol zmenený, a hlavne názov nového profilu (viď. Obrázok 17). Po kliknutí na notifikáciu sa otvorí hlavná obrazovka daného profilu.

V prípade, že užívateľ nemá nastavenú adresu pre profil Domov alebo Práca, je pomocou notifikácie upovedomený, aby si túto adresu pre správny chod aplikácie nastavil. (viď. Obrázok 16)



Obrázok 17: Notifikácia upozorňujúca na zmenu aktuálneho profilu na profil Podnik.



Obrázok 16: Notifikácia o chýbajúcej adrese pre profil Domov.

4.4 Systémové požiadavky

Základnou požiadavkou na beh aplikácie je zariadenie s operačným systémom Android. Konkrétne Android s minimálnym API levelom 17, čiže Android 4.2.x Jelly Bean. Avšak ako cieľová úroveň bol zvolený API level 23, čiže Android 6.0 Marshmallow. Dôvodom pre zvolenie minimálneho API levelu pre beh aplikácie bol fakt, že 87,3% zariadení s operačným systémom Android má API level 17 a vyšší [21]. Takto môže byť aplikácia nainštalovaná takmer na všetky zariadenia s operačným systémom Android. Pokiaľ by bol ako minimálny zvolený API level 19 (Android 4.4 KitKat), predstavovalo by to pokrytie len 73,8% zariadení, v prípade API levelu 21 (Android 5.0 Lollipop) dokonca len 40,4% [21].

Čo sa týka úložného priestoru, základná veľkosť aplikácie bez akýchkoľvek dát uložených užívateľom sa líši na základe verzie operačného systému. V prípade Androidu KitKat je to okolo 17 MB, na verzii Lollipop zaberá aplikácia okolo 25 MB a v prípade Androidu Marshmallow je to iba približne 9,5 MB. Je však potrebné si vyčleniť ešte ďalšie miesto pre dáta (napr. výnimky, časové predvoľby, užívateľom uložené príklady, atď.).

Pre správny chod aplikácie je taktiež potrebné, aby malo zariadenie GPS, mikrofón a senzory typu Accelerometer a Step Detector. Aplikácia bude bežať aj bez týchto senzorov, avšak nebude možné zabezpečiť získavanie potrebného kontextu. Týmto pádom nedokáže aplikácia určiť, v akom profile sa užívateľ nachádza, a teda stráca svoju hlavnú funkciu.

5 Testovanie

Na testovaní aplikácie sa zúčastnilo 20 používateľov. Vzorka používateľov obsahovala ľudí s rôznymi skúsenosťami so systémom Android a s aplikáciami vyvinutými pre tento systém. Od bežných „laických“ používateľov až po používateľov náročných a skúsených. Testovanie prebiehalo na zariadeniach s rozličnými verziami operačného systému Android. Konkrétne od verzie 4.3 Jelly Bean až po 6.0 Marshmallow.

Čo sa hardwarových špecifikácií týka, testovanie bolo vykonané na rôznych modeloch (low-end aj high-end) rozličných výrobcov, ako napríklad Samsung Galaxy Alpha, Samsung Galaxy A3 A300F, Samsung Galaxy S4 Mini, Google Nexus 7 2013, Sony Xperia Z3, Sony Xperia M4 Aqua, Sony Xperia SP, OnePlus X, Lenovo A7000, Asus Zenfone 5, Huawei Ascend G620s a ďalšie.

Na konci testovania vyplnili používatelia dotazník rozdelený do troch hlavných sekcií, a to grafické užívateľské rozhranie, praktickosť aplikácie a jej súčastí a funkčnosť aplikácie.

5.1 Vyhodnotenie dotazníkov

Pre potreby dotazníka sme využili Formuláre Google [23], ktoré umožňujú jednoduché vytváranie graficky prívetivých formulárov s rýchlym a ľahko šíriteľným zberom odpovedí. Formuláre Google taktiež poskytujú analýzu získaných formulárov v podobe grafov a výstup nazbieraných dát vo forme tabuliek.

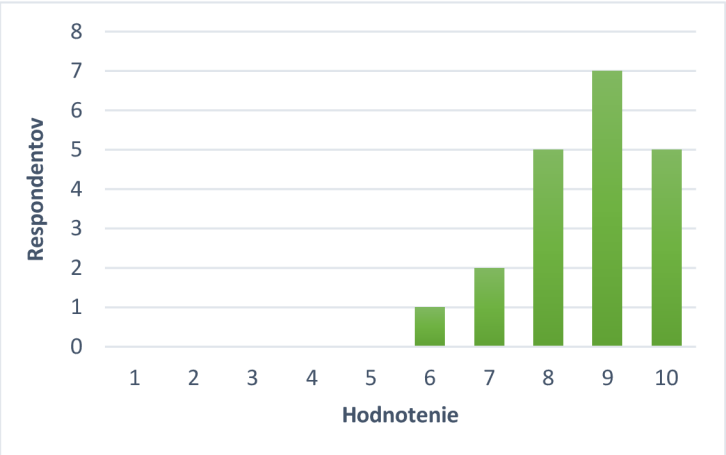
Ako už bolo spomenuté vyššie, v dotazníku užívatelia odpovedali na otázky z troch hlavných oblastí záujmu. V sérií otázok s Likertovou škálou [24] hodnotili grafické užívateľské rozhranie, praktickosť aplikácie a jej súčastí a funkčnosť aplikácie.

5.1.1 Grafické užívateľské rozhranie

V rámci sekcii o grafickom užívateľskom rozhraní odpovedali používatelia na tri otázky.

Ako hodnotíte prehľadnosť užívateľského prostredia?

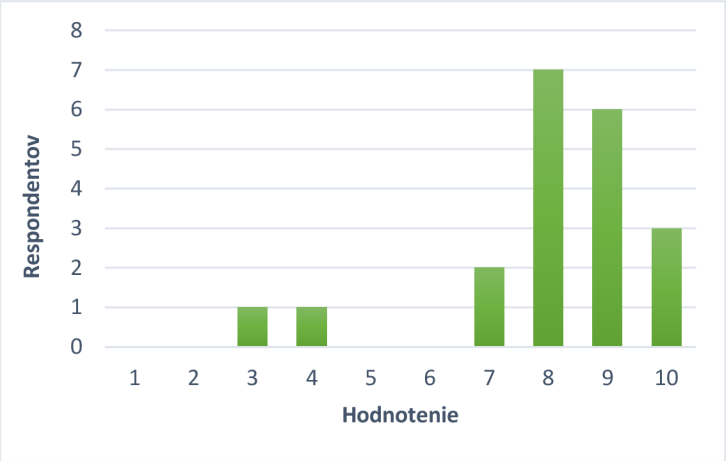
Užívatelia mohli na otázku odpovedať hodnotou na škále od 1 do 10, pričom 1 znamenalo „neprehľadné“ a 10 „prehľadné“. Čiže čím vyššie hodnotenie, tým prehľadnejšia podľa nich aplikácia bola. Z priloženého grafu možno vyčítať, že používatelia považujú aplikáciu za prehľadnú (viď. Obrázok 18). V priemere ohodnotili prehľadnosť užívateľského prostredia na 8,65 bodov z 10.



Obrázok 18: Graf hodnotení prehľadnosti užívateľského rozhrania.

Je používanie aplikácie intuitívne?

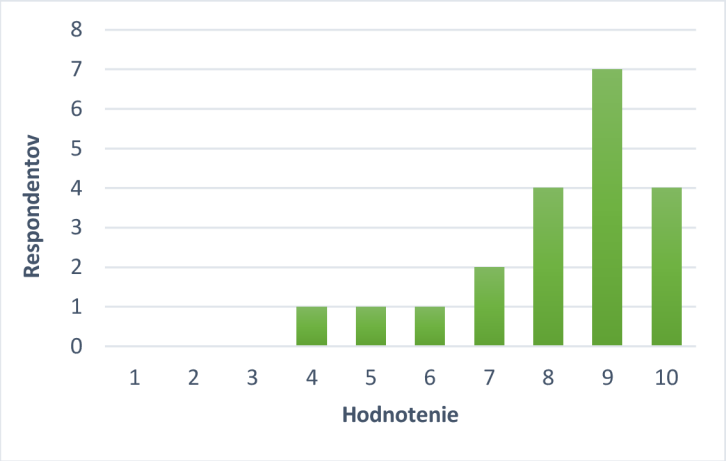
V ďalšej otázke používatelia hodnotili intuitívnosť aplikácie. Na otázku mohli odpovedať rovnako ako na predošlú, čiže hodnotením od 1 do 10, pričom hodnota 1 znamenala „vôbec“ a 10 „úplne“. Ako možno vidieť na grafe, našli sa aj respondenti, ktorí používanie aplikácie za veľmi intuitívne nepokladali, avšak väčšina užívateľov tvrdí opak (viď Obrázok 19). Priemerná hodnota užívateľských hodnotení je 8,05 bodov z 10.



Obrázok 19: Graf hodnotení intuitívnosti aplikácie.

Ako hodnotíte design aplikácie?

Užívatelia hodnotili taktiež design aplikácie. Hodnotiť mohli na škále od 1 („slabý“) po 10 („vynikajúci“). Ako možno vidieť na grafe, väčšine používateľov sa design aplikácie páčil (viď. Obrázok 20). V priemere ho respondenti ohodnotili na 8,2 bodov z 10.



Obrázok 20: Graf hodnotení designu aplikácie.

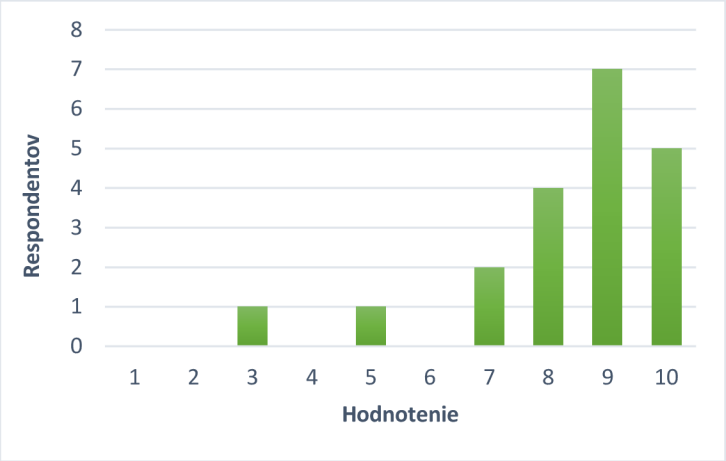
Na základe celkového priemeru hodnotení v tejto sekcii, 8,3 bodov z 10, možno predpokladať že testujúci používatelia boli s grafickým užívateľským rozhraním spokojní.

5.1.2 Praktickosť aplikácie a jej súčastí

V druhej sekcii odpovedali používatelia na otázky ohľadom praktického využitia aplikácie a jednotlivých jej súčastí. Otázky boli tri a to nasledovné:

Ako hodnotíte praktickosť používania aplikácie?

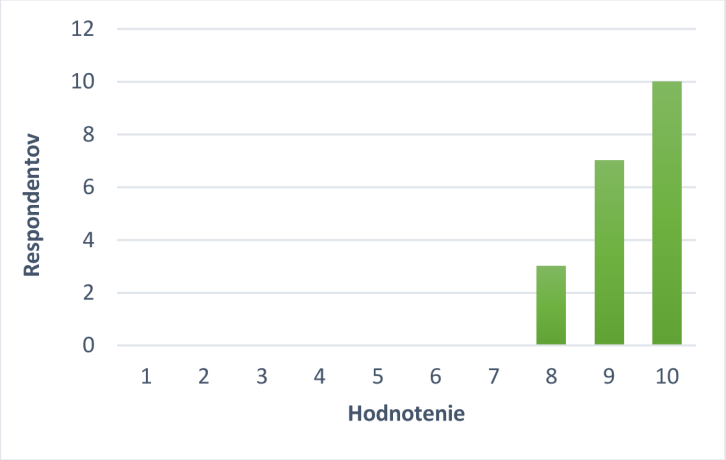
Praktickosť používania aplikácie mohli užívatelia ohodnotiť na škále od 1 („nepraktické“) po 10 („praktické“). Na priloženom grafe so zozbieranými odpoveďami je možné vidieť, že väčšina užívateľov považuje praktickosť aplikácie za vysokú (viď. Obrázok 21). V priemere užívatelia udelili praktickosti používania tejto aplikácie 8,35 bodov z 10.



Obrázok 21: Graf hodnotení praktickosti používania aplikácie.

Ako hodnotíte praktickosť možnosti pridať výnimky pre každý profil?

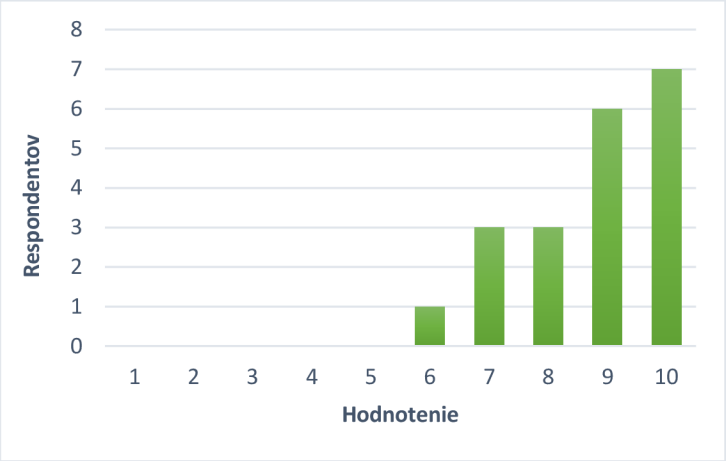
V ďalšej otázke hodnotili užívatelia praktickosť možnosti pridávania výnimiek. Znova mali možnosť hodnotiť na stupnici od 1 („nepraktické“) po 10 („praktické“). Ako vidno na priloženom grafe, užívatelia považujú túto súčasť aplikácie za veľmi praktickú (viď. Obrázok 22). V priemere udelili praktickosti tejto možnosti 9,35 bodov z 10.



Obrázok 22: Graf hodnotení praktickosti možnosti pridať výnimky pre každý profil.

Ako hodnotíte praktickosť možnosti zadefinovať si časové predvoľby?

Testujúci užívatelia mali taktiež možnosť ohodnotiť praktickosť časových predvolieb. Učiniť tak mohli hodnotením na škále od 1 („nepraktické“) po 10 („praktické“). Z priloženého grafu vyplýva, že užívatelia túto možnosť hodnotia ako praktickú (viď. Obrázok 23). Priemerné hodnotenie praktickosti možnosti zadefinovať si časové predvoľby je 8,75 bodov z 10.



Obrázok 23: Graf hodnotení praktickosti možnosti zadefinovať si časové predvoľby.

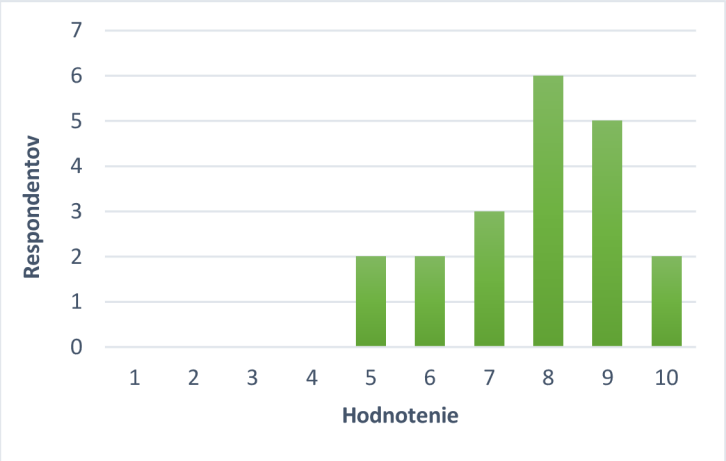
Celkový priemer hodnotení v tejto sekcii je 8,82 bodov z 10. Na základe tohto údaju je možné predpokladať, že z hľadiska praktickosti boli testujúci užívatelia s vyvinutou aplikáciou vysoko spokojní. Taktiež je vidno, že užívatelia oceňujú viac možnosť pridať si výnimky pre každý profil ako možnosť zadefinovať si časové predvoľby.

5.1.3 Funkčnosť aplikácie

Posledná časť dotazníka sa týka funkčnosti testovanej aplikácie. V tejto časti boli respondentom taktiež položené tri otázky.

Ako hodnotíte presnosť určovania aktuálneho profilu?

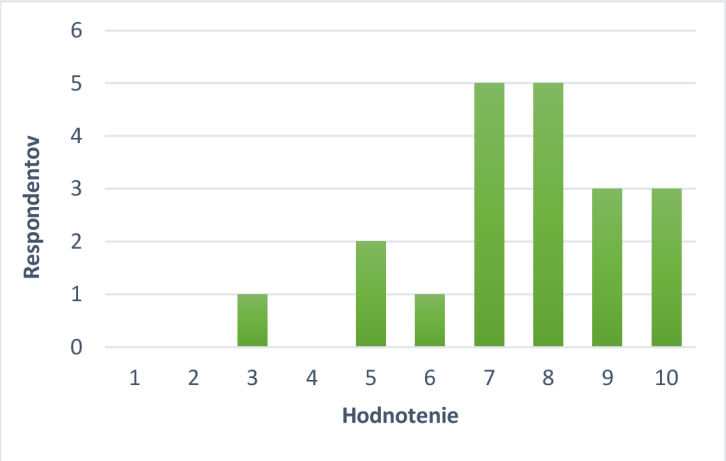
Prvá otázka tejto sekcie sa týka presnosti určovania aktuálneho profilu. Ide teda o to, či boli užívatelia spokojní s tým, ako aplikácia vyhodnocovala získaný kontext a určovala aktuálny profil. Ich skúsenosti mohli premietnuť do hodnotenia na škále od 1 („nepresné“) po 10 („presné“). Konkrétne výsledky tejto otázky je sa nachádzajú v grafe nižšie (viď. Obrázok 24). V priemere používatelia ohodnotili presnosť určovania aktuálneho profilu na 7,8 bodov z 10.



Obrázok 24: Graf hodnotení presnosti určovania aktuálneho profilu.

Ako hodnotíte spotrebu batérie?

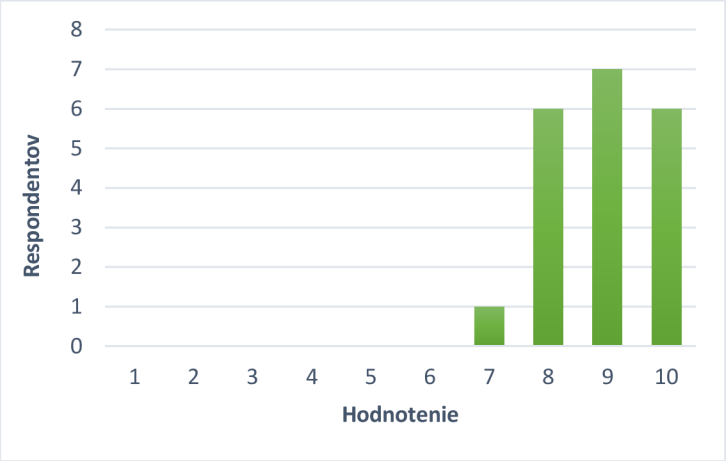
V ďalšej otázke mali testujúci možnosť ohodnotiť vplyv aplikácie na spotrebu batérie. Hodnotiť mohli na škále od 1 po 10, pričom 1 predstavuje nevyhovujúcu spotrebu a 10 vyhovujúcu spotrebu. Z grafu možno vyčítať, že používatelia pocítili vplyv práce so senzormi, GPS a neustáleho behu na pozadí na spotrebu batérie (viď. Obrázok 25). Stále však považujú spotrebu batérie za vyhovujúcu a udelili jej v priemere 7,55 bodov z 10.



Obrázok 25: Graf hodnotení spotreby batérie

Ako hodnotíte celkovú funkčnosť aplikácie?

Užívatelia mali na záver dotazníka možnosť ohodnotiť celkovú funkčnosť aplikácie. Hodnotiť mohli, tak ako predtým, na škále od 1 po 10. Pričom 1 predstavuje slabú funkčnosť a 10 vynikajúcu funkčnosť. Z ich hodnotení sa dá vyčítať, že funkčnosť testovanej aplikácie sa blíži k vynikajúcej (viď. Obrázok 26). V priemere ohodnotili celkovú funkčnosť aplikácie na 8,9 bodov z 10.



Obrázok 26: Graf hodnotení celkovej funkčnosti aplikácie.

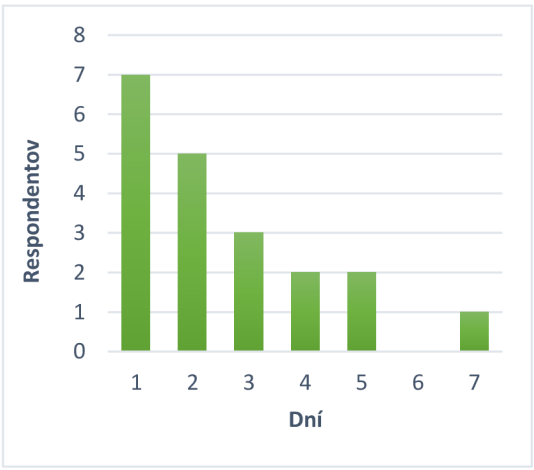
5.1.4 Štatistické údaje

Zo štatistického hľadiska som sa pýtal respondentov na ich vek, počas akej doby aplikáciu testovali a na to, či majú skúsenosti s vývojom pre operačný systém Android.

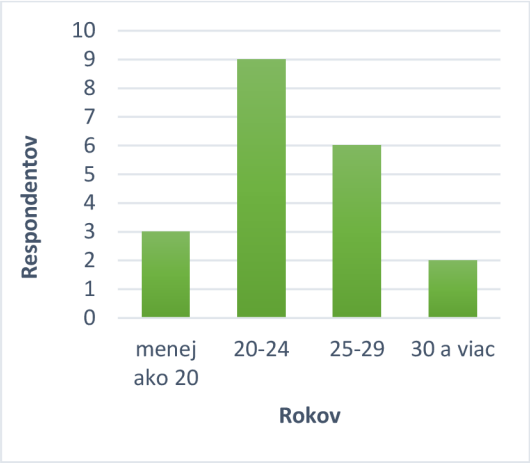
Vek respondentov je znázornený v priloženom grafe (viď. Obrázok 28). Najviac používateľov bolo z vekovej kategórie od 20 do 24 rokov.

V priemere testovali používatelia aplikáciu 2,55 dňa. Konkrétne počty dní, počas ktorých respondenti aplikáciu testovali, možno vidieť v priloženom grafe (viď. Obrázok 27).

Väčšina testujúcich užívateľov, presne 75%, nemala predošlé skúsenosti s vývojom pre operačný systém Android.



Obrázok 27: Graf zobrazujúci koľko respondentov testovalo aplikáciu akú dlhú dobu.



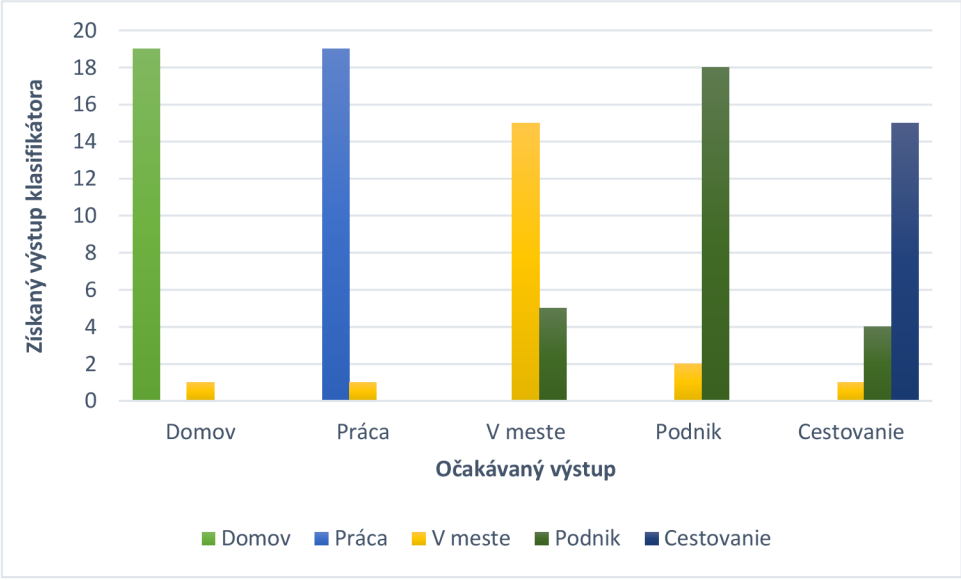
Obrázok 28: Graf zobrazujúci koľko respondentov patrilo do ktorej vekovej kategórie.

5.2 Vyhodnotenie umelého testovania

Klasifikátor a jeho schopnosť určiť zo získaného kontextu profil, v ktorom sa používateľ nachádza, som podrobil aj umelému testovaniu. Toto testovanie spočívalo v tom, že na vstup klasifikátora boli posielané reťazce kontextových dát, ktoré bolo treba vyhodnotiť. Správnym výsledkom bolo, pokiaľ klasifikátor priradil k týmto dátam takú triedu/profil, aká k nim prináležala. Takýmto testovaním je možné overiť správnosť tréningovej sady, z ktorej klasifikátor vychádza.

Pri testovaní bola použitá tréningová sada v rozsahu 500 klasifikácií, čiže 100 klasifikácií pre každý profil. Testovaná tréningová sada v nezmenenej podobe predstavuje aj tréningovú sadu výslednej aplikácie.

Testovacia sada obsahovala 100 klasifikácii, pre každý profil 20. Počty správne a nesprávne vyhodnotených klasifikácii pre každý profil možno vidieť v priloženom grafe (viď. Obrázok 29). Celková úspešnosť klasifikátora na tejto testovacej sade bola 86%. Avšak výsledky takéhoto testovania treba brať s rezervou. Pri skutočnom používaní aplikácie môžu nastať komplikácie, s ktorými sa tu nepočíta (napr. chybné dáta zo senzorov, slabý signál GPS, užívateľ nemá bežný časový rozvrh, atď.).



Obrázok 29: Graf zobrazujúci výstupy klasifikátora pre testovaciu sadu.

V prípade kontextových dát, ktoré mali byť klasifikované ako profil Domov, dosiahol klasifikátor 95% úspešnosť. Iba v jednom prípade vyhodnotil klasifikátor kontext nesprávne. Hlavným dôvodom tejto chyby bola rýchlosť používateľa. V príklade na vstupe bola zadaná rýchlosť cez 1,5 m/s. Pre túto rýchlosť vypočítal klasifikátor v profile Domov veľmi nízku pravdepodobnosť, ktorá mala za následok to, že pravdepodobnosť pre profil V meste bola vyššia ako pre profil Domov.

Úspešnosť pre profil Práca bola taktiež 95% a dôvodom pochybenia bola, tak ako aj v prípade profilu Domov, hlavne zvýšená rýchlosť užívateľa.

Ten istý problém sa vyskytol aj v prípade profilu Podnik. V dvoch prípadoch klasifikátor priradil kontextovým dátam profil V meste. Celková úspešnosť teda bola 90%.

Profil V meste vyhodnotil klasifikátor správne v 75% prípadov. Najväčšiu chybovosť vykazoval, keď bolo zariadenie v polohe ležmo, rýchlosť bola nulová a hlučnosť vysoká. Na základe týchto, ale aj iných, aspektov klasifikátoru vyšla pravdepodobnosť, že sa užívateľ nachádza v klube vyššia ako pre profil V meste.

Úspešnosť klasifikátora v príkladoch pre profil Cestovanie bola 75%. Keď bola zadaná príliš nízka alebo nulová rýchlosť, klasifikátor priradil ku kontextovým dátam triedu/profil V meste alebo Podnik.

5.3 Zhrnutie výsledkov

Výsledná aplikácia zostavuje kontext z údajov získaných pomocou senzorov zariadenia (poloha zariadenia, úroveň hluku, orientácia zariadenia, chôdza užívateľa a rýchlosť pohybu) a údajov získaných zo systému (čas a deň v týždni). Na základe týchto získaných údajov môže klasifikátor, postavený na princípe Naïve Bayes klasifikátora, v aplikácii vyhodnotiť, s akou pravdepodobnosťou sa užívateľ nachádza v tom-ktorom profile. Konkrétne sa jedná o profily Domov, Práca, V meste, Podnik a Cestovanie. Užívateľ si pre každý z týchto profilov môže navoliť samostatné zvukové nastavenie. Pokiaľ si napríklad navolí, že v profile Domov má byť telefón v tichom režime, a klasifikátor vyhodnotí na základe získaného kontextu, že užívateľ je doma, tak sa telefón automaticky prepne do tichého režimu. O zmene aktuálneho profilu je užívateľ informovaný pomocou notifikácií. Ako sa ukázalo pri testovaní, používanie takejto aplikácie považuje väčšina užívateľov za veľmi praktické.

Súčasťou aplikácie sú aj možnosti pridať si výnimky pre každý profil a nastaviť si časové predvoľby. Výnimky fungujú tak, ako ich názov napovedá. Používateľ si v konkrétnom profile vyberie niekoho zo svojich kontaktov a pridelí mu osobitné nastavenie zvuku. Následne sa v prípade hovoru alebo správy od dotyčnej osoby uplatní toto osobitné nastavenie bez ohľadu na zvukové nastavenie profilu. Túto možnosť môže užívateľ uplatniť napríklad pokiaľ nechce byť rušený, ale existuje osoba alebo osoby, ktoré sa mu aj v takomto prípade musia dokázať dovolať. Ako ukázali výsledky testovania, väčšina užívateľov považuje túto možnosť za veľmi praktickú.

Časové predvoľby spočívajú v tom, že si užívateľ určí istý časový úsek, počas ktorého má platiť isté nastavenie zvuku. Užívateľ si môže zvoliť aj dni v týždni, počas ktorých sa táto predvoľba môže opakovať. Časové predvoľby sú nadradené profilom. To znamená, že pokiaľ práve niektorá platí, aplikácia nehľadí na aktuálny profil, ale uplatňuje nastavenia danej predvoľby. V testovaní užívatelia taktiež ohodnotili túto súčasť aplikácie ako veľmi praktickú, avšak nie až natoľko ako výnimky.

Nemalú súčasť výslednej aplikácie predstavuje grafické užívateľské rozhranie. Toto rozhranie bolo navrhnuté tak, aby bolo užívateľsky prívetivé, prehľadné a aby bola vďaka nemu práca s aplikáciou intuitívna. Ako ukazujú výsledky užívateľských testov, tak sa tento cieľ vo výslednej aplikácii podarilo naplniť.

Problémom sa počas vývoja ukázala byť spotreba batérie. Pri neustálom získavaní dát zo senzorov a vyhodnocovaní týchto dát spotreba batérie výrazne stúpla. Po následnej optimalizácii

a navýšení intervalu medzi jednotlivými získavaniami dát zo senzorov sa podarilo znížiť spotrebu batérie oproti základnej verzii o viac ako 75%. Výslednú spotrebu považujú za prijateľnú aj užívatelia, ktorí aplikáciu testovali.

Keď sa zameriame na presnosť určovania profilu a celkovú funkčnosť aplikácie, tak aj na základe výsledkov užívateľských testov môžem prehlásiť, že sa podarilo naplniť ciele tejto práce. Užívateľ má dokonca možnosť aplikáciu učiť, a tým ešte zvýšiť presnosť určovania profilov pre jeho osobné potreby.

6 Záver

Cieľom tejto práce bolo vytvoriť aplikáciu pre zariadenia používajúce operačný systém Android, ktorá na základe získaného kontextu vyhodnotí, kde sa používateľ nachádza, a aplikuje ním preddefinované zvukové nastavenia pre danú situáciu. Týmto mu umožní automatizovať prácu so zariadením, ktorú inak musel sám vykonávať. Taktiež sa môže vyhnúť situáciám, keď si zvukové nastavenia zabudol upraviť a následne sa zariadenie vyzváňaním postaralo o neželanú situáciu, alebo naopak sa mu v potrebných chvíľach nemohol niekto dovolať.

Aplikácia bola úspešne navrhnutá, implementovaná a otestovaná. Na užívateľskom testovaní sa podieľalo 20 užívateľov. V dotazníku, ktorý respondenti po otestovaní aplikácie vyplnili, sa hodnotenia pohybovali v priemere na úrovni osem a pol bodu z maximálneho počtu desať. Schopnosť správne rozoznať aktuálny profil bola taktiež overená aj umelým testovaním na testovacej sade v rozsahu 100 príkladov. Celková úspešnosť, ktorú klasifikátor v tomto testovaní dosiahol, je 86%.

Prínos tejto práce pre mňa predstavujú získané skúsenosti s vývojom aplikácie, od návrhu až po testovanie, od procesov na pozadí až po užívateľské rozhranie. Taktiež som sa vďaka tejto práci zoznámil s vývojom pre Android platformu aj s operačným systémom samotným, nakoľko som s vývojom pre tento systém doposiaľ nemal žiadne skúsenosti.

V budúcnosti by som sa chcel pri vývoji tejto aplikácie zamerať na zvýšenie presnosti pri určovaní profilov a rozšíreniu funkčnosti. Pod rozšírením funkčnosti by som si vedel predstaviť aj možnosť blokovania hovorov a správ, možnosť nastaviť výnimky spomedzi aplikácií a možnosť nastaviť výnimky v prípade časových predvolieb. Taktiež by som chcel v budúcnosti implementovať nastavenia, v ktorých by si užívateľ mohol prispôsobiť napríklad interval získavania a vyhodnocovania kontextu (mohol by tým ovplyvňovať spotrebu batérie na úkor aktuálnosti profilu), či citlivosť na polohu v prípade profilu Domov a Práca.

Literatúra

[1] Context: Definition of Context by Merriam-Webster. In: *Merriam-Webster* [online]. Encyclopædia Britannica [cit. 2016-01-21]. Dostupné z: http://www.merriam-webster.com/dictionary/context

[2] Context. In: *Wiktionary* [online]. 2016 [cit. 2016-01-21]. Dostupné z: https://en.wiktionary.org/wiki/context

[3] PREKOP, Paul a Mark BURNETT. Activities, context and ubiquitous computing. *Computer Communications* [online]. 2003, **26**(11), 1168-1176 [cit. 2016-01-21]. DOI: 10.1016/S0140-3664(02)00251-7. ISSN 01403664.

[4] EMMANOUILIDIS, Christos, Remous-Aris KOUTSIAMANIS a Aimilia TASIDOU. Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications* [online]. 2013, **36**(1), 103-125 [cit. 2016-01-21]. DOI: 10.1016/j.jnca.2012.04.007. ISSN 10848045.

[5] SCHILIT, B., N. ADAMS a R. WANT. Context-Aware Computing Applications. In: *1994 First Workshop on Mobile Computing Systems and Applications* [online]. IEEE, 1994, s. 85-90 [cit. 2016-01-21]. DOI: 10.1109/WMCSA.1994.16. ISBN 978-0-7695-3451-0.

[6] SCHILIT, B., N. ADAMS a R. WANT. Context-Aware Computing Applications. In: *1994 First Workshop on Mobile Computing Systems and Applications* [online]. IEEE, 1994, s. 85-90 [cit. 2016-01-21]. DOI: 10.1109/WMCSA.1994.16. ISBN 978-0-7695-3451-0.

[7] DEY, Anind K. *Providing Architectural Support for Building Context-Aware Applications*. Georgia, 1999. Disertační práce. Georgia Institute of Technology.

[8] WEI LIU, XUE LI a DAOLI HUANG. A survey on context awareness. In:*2011 International Conference on Computer Science and Service System (CSSS)* [online]. IEEE, 2011, s. 144-147 [cit. 2016-01-21]. DOI: 10.1109/CSSS.2011.5972040. ISBN 978-1-4244-9762-1.

[9] MOSTEFAOUI, G.K., J. PASQUIER-ROCHA a P. BREZILLON. Context-Aware Computing: A Guide for the Pervasive Computing Community. In: *The IEEE/ACS International Conference on Pervasive Services*[online]. IEEE, 2004, s. 39-48 [cit. 2016-01-21]. DOI: 10.1109/PERSER.2004.14. ISBN 0-7695-2535-0.

[10] HAN, Jiawei a Micheline KAMBER. *Data mining: concepts and techniques*. San Francisco: Morgan Kaufmann Publishers, c2001. Morgan Kaufmann series in data management systems. ISBN 1-55860-489-8.

[11] FERNÁNDEZ, Lourdes Nicolás. *Context Aware Application for Android*. Brno, 2012. Diplomová práce. Vysoké učení technické v Brně.

[12] KOVÁČOVÁ, Alžbeta a Lucia BRÉDOVÁ. Naive Bayes. In: *Objavovanie znalostí* [online]. 2012 [cit. 2016-02-04]. Dostupné z: http://oz.koncz.sk/index.php/operatory-rapidminer/80-naive-bayes

[13] KORDÍK, Pavel a Jan MOTL. Vytěžování znalostí z dat. Přednáška 7: Bayesovská klasifikace. In: *Edux.fit.cvut.cz* [online]. 2011 [cit. 2016-02-04]. Dostupné z: https://edux.fit.cvut.cz/oppa/BI-VZD/prednasky/p7-Bayes.pdf

[14] Data Mining - Decision Tree Induction. In: *Tutorialspoint* [online]. 2016 [cit. 2016-03-10]. Dostupné z: http://www.tutorialspoint.com/data_mining/dm_dti.htm

[15] NEDELCU, Aleandru. How To Build a Naive Bayes Classifier. In: *Bionic Spirit* [online]. 2012 [cit. 2016-03-15]. Dostupné z: https://bionicspirit.com/blog/2012/02/09/howto-build-naive-bayes-classifier.html

[16] *Google Developers* [online]. [cit. 2016-03-20]. Dostupné z: https://developers.google.com/

[17] *Android Developer* [online]. [cit. 2016-03-20]. Dostupné z: http://developer.android.com/index.html

[18] Event Handling Guide for iOS. In: *IOS Developer Library* [online]. [cit. 2016-03-20]. Dostupné z: https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/motion_event_basics/motion_event_basics.html#//apple_ref/doc/uid/TP40009541-CH6-SW14

[19] *Realm* [online]. [cit. 2016-03-23]. Dostupné z: https://realm.io/

[20] *SQLite* [online]. [cit. 2016-03-23] Dostupné z: https://www.sqlite.org/

[21] Dashboards: Platform Versions. In: *Android Developer* [online]. [cit. 2016-04-04]. Dostupné z: https://developer.android.com/about/dashboards/index.html

[22] Android Studio Overview. In: *Android Developer* [online]. [cit. 2016-03-27]. Dostupné z: https://developer.android.com/about/dashboards/index.html

[23] *Formuláre Google* [online]. [cit. 2016-04-07]. Dostupné z: https://www.google.sk/intl/sk/forms/about/

[24] LIKERT, Rensis. A technique for the measurement of attitudes. *Archives of psychology*. New York: The Science Press, 1932, **22**(140), 5-55.