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

Bakalářská práce

Použití Android zařízení v roli alarmu

Plzeň 2017 Jakub Sobek

Místo této strany bude zadání práce.

Prohlášení

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

V Plzni dne 28. června 2017

Jakub Sobek

Poděkování

Rád bych poděkoval panu Ing. Ladislavu Pešičkovi, za ochotu, vstřícnost, odborné vedení, rady a připomínky při vytváření této bakalářské práce.

Abstract

The aim of this bachelor thesis is creation of a security alarm application which uses sensors of mobile phones with an operating system Android. This application is designed for this kind of mobile phones. First part of the thesis is dedicated to the operating system Android and to its integrated development environment called Android Studio, in which the application is developed. Next part is focused on the most important sensors that are mostly used for similar applications. These are motion sensors, environmental sensors and position sensors. Then analysis of the similar application is made with the focus on the basic description and function. Last part of the thesis follows description of the development of the security alarm application, its function and testing.

Abstrakt

Cílem této bakalářské práce je vytvoření aplikace bezpečnostního alarmu, která využívá senzorů mobilních telefonů s operačním systémem Android. Pro tyto mobilní telefony je vytvářená aplikace určená. První část práce se věnuje operačnímu systému Android a jeho vývojovému prostředí s názvem Android studio, ve kterém je uvedená aplikace vyvíjena. Další část je poté zaměřena na nejdůležitější senzory, jež jsou nejčastěji pro obdobné aplikace využívány. Jedná se o pohybové senzory, senzory prostředí a senzory polohy. Následně je proveden rozbor obdobných aplikací se zaměřením na jejich základní popis a funkce. V poslední části práce následuje popis samotné tvorby aplikace bezpečnostního alarmu, její funkcionality a testování.

Obsah

1 Úvod

Cílem této bakalářské práce je vytvoření aplikace bezpečnostního alarmu pro mobilní telefony s operačním systémem Android, o kterém bude stručně pojednáno v druhé kapitole.

Úkolem této aplikace pak bude využívat senzorů mobilních zařízení pro detekci různých změn prostředí a poté informovat uživatele o dané změně na základě určených prahů citlivosti zmíněných senzorů. Z uvedených důvodů bude v této bakalářské práci věnován prostor popisu nejdůležitějších senzorů, které se při tvorbě obdobných aplikací využívají.

Aby se vyvíjená aplikace bezpečnostního alarmu vymezila oproti již existujícím aplikacím, budou vybrány nejvíce podobné aplikace, a to ke zkoumání jejich funkcionalit. Výsledným produktem této bakalářské práce pak bude bezpečnostní alarm, který bude využívat určitých senzorů mobilních zařízení a jehož výhody a nevýhody oproti stávajícím aplikacím budou shrnuty v závěru.

2 Operační systém Android

Jak již naznačuje název této bakalářské práce, bude aplikace alarmu určena pro jeden z nejpoužívanějších operačních systémů pro mobilní zařízení, a to pro operační systém Android. Je tedy pro začátek nutné, seznámit se s touto platformou. Následně bude věnována pozornost vývojovému prostředí, v němž byla aplikace bezpečnostního alarmu vyvíjená.

2.1 Obecná charakteristika

Tento operační systém vytvořený společností Android, Inc. v roce 2003 v Palo Alto v Kalifornii ve Spojených státech amerických, patří v současnosti u mobilních zařízení mezi jeden z nejrozšířenějších. Například v roce 2016 dominoval operační systém Android na trhu chytrých telefonů podílem 86,8%.[15]

Android je rozsáhlý operační systém, který vlastní v současné době společnost Google. Ta v srpnu 2005 společnost Android, Inc. odkoupila a vytvořila z ní svou dceřinou společnost.

Jedná se o operační systém založený na *open source* platformě. To znamená, že se jedná o počítačový software s otevřeným zdrojovým kódem, což sebou přináší snadnou dostupnost, a to jak technickou, tak licenční. Uživatel tudíž může při splnění určitých podmínek, využívat systém zadarmo. Je mu také umožněn přístup ke zdrojovým kódům, které může následně používat a dokonce dále upravovat.[20] Tato platforma vznikla především pro mobilní zařízení jako jsou například chytré telefony, PDA, navigace a tablety.

U chytrých telefonů umožňuje vývojářům vytvářet mobilní aplikace, které se mohou týkat například obsluhy telefonních hovorů, posílání textových zpráv či využívání fotoaparátu.[20] Takovouto aplikací je samozřejmě i aplikace bezpečnostního alarmu, která je v rámci této bakalářské práce vytvořena. Android se však stále vyvíjí a stále rozšiřuje své možnosti, takže ho dnes můžeme objevit i u hodinek a chytrých televizí. Do budoucna ho nalezneme i v autech, zábavních systémech letadel a dokonce i v robotech.[19]

Android je založen na Linuxovém jádře různých verzí, jehož úkolem je zabezpečení systému jako celku, správa paměti, správa procesů, přístup k síti a

ovladačům všech vnitřních senzorů a komponent. Důležité je však upozornit, že jednotlivé aplikace k funkcím jádra nepřistupují přímo, ale přistupují k němu prostřednictvím Android API - *application programming interference*, v češtině označené jako rozhraní pro programování aplikací.[20] Android nabízí bohatý aplikační *framework*, jenž umožňuje vytvářet inovativní aplikace a hry pro mobilní zařízení v jazykovém prostředí Java. Aplikace Androidu jsou tvořeny jako kombinace rozlišných komponent, jež mohou být uplatněny individuálně. Například některá individuální *aktivita* zajišťuje jednu obrazovku uživatelského rozhraní, a *služba* nezávisle na této obrazovce vykonává svou práci na pozadí.[6] Zde se nabízí se otázka, jaký je rozdíl mezi aktivitami a službami. Aktivity jsou jakýmisi stavebními bloky uživatelského rozhraní s krátkým životním cyklem, jež můžeme kdykoli ukončit. Služby jsou naopak navrženy tak, aby byly v neustálém provozu, a pokud je to nutné, aby byly v provozu nezávisle na aktivitách. Příkladem služby může být přehrávaní hudby na pozadí, a to i když jeho řídící aktivita již neběží.[19]

Aplikace, která je výsledkem této bakalářské práce, je kompatibilní se všemi verzemi Androidu od verze Android 4.0.3 Ice Cream Sandwich, jehož API level je 15. Pro naprogramování aplikace bezpečnostního alarmu byla významná aktualizace, která se objevila u verze Android 6.0 (API level 23). Od této aktualizace uživatelé dávají oprávnění aplikacím za jejich běhu, a nikoli ve chvíli kdy tuto aplikaci instalují. Tento přístup zjednodušuje instalační proces aplikace, neboť uživatelé nemusí dávat oprávnění, když instalují nebo aktualizují aplikaci. Tato aktualizace také poskytuje uživateli větší kontrolu nad funkcionalitou aplikace. Například si uživatel může vybrat, že dá aplikaci fotoaparátu přístup k fotoaparátu ale nikoli již k poloze zařízení. Uživatel může kdykoli odvolat své svolení tím, že půjde do nastavení aplikace.[11]

Systém oprávnění je rozdělen do dvou skupin: obvyklé (*normal*) a nebezpečné (*dangerous*). Obvyklé oprávnění přímo neohrožuje soukromí uživatele. Pokud aplikace obsahuje ve svém prohlášení obvyklé oprávnění, systém dává svolení automaticky. Nebezpečné svolení může dávat aplikaci přístup k důvěrným informacím uživatele. Pokud aplikace obsahuje nebezpečné oprávnění, uživatel musí této aplikaci dát souhlas výslovně.[11]

2.2 Android Studio

V roce 2013 přišel Google s novým vývojovým prostředí, a to Android Studiem. Jedná se o oficiální *Integrated Development Environment* (IDE) - integrované vývojové prostředí pro vývoj Android aplikací založený na *IntelliJ IDEA*. Android Studio například obsahuje flexibilní systém pro sestavování programů založený na nástroji *Gradle*, rychlý a na nástroje bohatý emulátor, jednotné prostředí, kde lze vyvíjet pro všechna zařízení Androidu, *Instant Run* k aplikování změn bez potřeby znovustavění celého APK, šablony kódů a integrace s GitHubem k napomáhání vyvíjení běžných aplikačních funkcí, rozsáhlé nástroje testování a fameworků, podpora C++ a NDK, zabudovaná podpora Google Cloud Platformy, atd.[8] Nejnovější verzí Android studia je v současné době verze Android Studia v2.3.3, která vyšla v červnu 2017.

Touto stručnou exkurzí o operačním systému Android je ukončena druhá kapitola této bakalářské práce. Následující kapitola se bude týkat dalšího významného tématu, a to tématu senzorů, které můžeme na mobilních zařízení s operačním systémem Android najít.

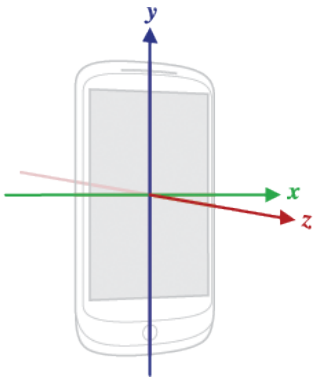
3 Senzory mobilních zařízení s operačním systémem Android

K tomu aby bylo možné vytvořit aplikaci bezpečnostního alarmu, je potřeba se zaměřit na „smysly“ - senzory, kterými je vybavena většina chytrých telefonů a díky kterým je možné takovouto aplikaci v praxi využít. Každý chytrý telefon či jiné zařízení s operačním systémem Android může obsahovat různé senzory. Některá zařízení mají takové senzory, které jinému zařízení mohou chybět a obráceně. Tato kapitola bude zaměřena na nejdůležitější senzory, které obsahují téměř všechny chytré telefony s platformou Android. Na konci této kapitoly bude poté vysvětleno, proč byly některé z těchto senzorů pro aplikaci bezpečnostního alarmu vybrány a proč jiné nikoli.

Operační systém Android podporuje tři velké kategorie senzorů:

- *motion sensors* - pohybové senzory
- *environmental sensors* - senzory prostředí
- *position sensors* - senzory polohy

Pohybové senzory měří síly zrychlení a rotační síly podél tří os. Tato kategorie zahrnuje akcelerometr, senzor gravitace, gyroskop a rotační vektorové senzory. Skupina senzorů prostředí měří různé parametry prostředí kolem daného zařízení jako je například pokojová teplota a tlak, vlhkost a osvětlení. Tato skupina senzorů obsahuje barometry, teploměry a fotometry. Poslední kategorií jsou senzory polohy. Tyto senzory měří fyzickou polohu zařízení. Do této kategorie patří senzory orientace a magnetometry.[13] Senzory používají tří- osí koordinační systém k vyjádření dat. Pro mnoho senzorů je koordinační systém definován vzhledem k obrazovce zařízení, ve chvíli kdy je zařízení ve své základní poloze.Na obrázku je vidět koordinační systém, který je používán API pro senzory.[12](viz Obrázek 3.1)



Obrázek 3.1: Koordinační systém používaný API pro senzory[12]

Koordinační systém telefonu je založen na obrazovce a na výchozí orientaci telefonu. Osy x, y, z pracují tak, že:

- osa x- je osa horizontální s pozitivními hodnotami napravo a s negativními hodnotami nalevo
- osa y- je osa vertikální s pozitivními hodnotami nahoře a s negativními hodnotami dole
- osa z- pozitivní hodnoty vycházejí z obrazovky směrem dopředu a negativní hodnoty směrem za obrazovku tak, že bod O zůstává na obrazovce.[21]

Důležité je pochopit, že když se mění orientace obrazovky zařízení, tak se osy nemění. To znamená, že koordinační systém senzorů se pohybem zařízení nikdy nemění.[12]

Nyní lze přistoupit k popisu jednotlivých senzorů, které jsou pro tuto bakalářskou práci z hlediska tvorby aplikace bezpečnostního alarmu důležité. Jedná se ze skupiny pohybových senzorů o akcelerometr a gyroskop, dále ze skupiny senzorů prostředí o senzor intenzity okolního osvětlení, a z kategorie senzorů polohy o geolokační senzor a senzor přiblížení. Mimo tyto kategorie budou nakonec rozebrány vlastnosti fotoaparátu a mikrofonu.

3.1 Pohybové senzory

Operační systém Android poskytuje několik senzorů, které sledují pohyb zařízení. Dle typu senzoru se pak liší možná senzorová architektura: senzory

gravitace, lineárního zrychlení či krokoměru jsou více proměnlivé, neboť mohou být buď hardwarově založené nebo softwarově založené Naopak senzory akcelometr a gyroskop jsou vždy hardwarově založené.[9]

3.1.1 Akcelerometr

Akcelerometr, je jedním ze základních senzorů, kterým je vybavena většina chytrých telefonů a který se řadí mezi pohybové senzory, jenž jsou hardwarově založené.

Takové senzory jsou hmotné komponenty, které jsou vestavěné do přenosných zařízení a tabletů, a která čerpají svá data tak, že přímo měří určité vlastnosti prostředí jako je například zrychlení, intenzita geomagnetického pole či změna úhlů.[13]

Akcelerometr, jak již naznačuje název, měří zrychlení zařízení, a to v m^2/s. Tento senzor vypočítává zrychlení ze všech tří fyzických os zařízení - x, y a z, kdy samozřejmě do výpočtu zahrnuje i sílu gravitace.[13] Měří tedy zrychlení aplikované na zařízení zahrnujíc gravitaci.[9] Tento senzor je navržen tak, aby při změně z konstantní nebo z nulové rychlosti zaregistroval tuto změnu.[14] Jednoduše lze říci, že v momentě kdy například telefon leží v klidu na stole, působí na něho pouze gravitační zrychlení země - g - s hodnotou 9, 81 m^2/s. Ve chvíli, kdy však telefon zvedneme, získá telefon již jiné zrychlení, které poté akcelerometr ze zmíněných tří os vypočítává a tato data poté poskytuje jednotlivým aplikacím. Jeho využití je i u her, kdy máme například nějaké těleso udržet pomocí natáčení zařízení na dráze.

Na závěr lze zmínit, že akcelerometr využívá piezoelektirický jev, tedy mikroskopické krystaly, na kterých se při působení vibrací vytváří napětí, které odpovídá určitému zrychlení.[14]

3.1.2 Gyroskop

Také gyroskop nalezneme v mnoha zařízení s operačním systémem Android. I tento senzor se řadí mezi pohybové senzory, které jsou hardwarově založené.

Gyroskop, který se často připojuje v mobilních zařízení k akcelerometru, měří úhlovou rychlost kolem všech tří fyzických os, a to v rad/s. Jeho cílem je tedy detekce otáčení zařízení jako například zatočení, natočení zařízení, apod.[13] Důvodem proč je gyroskop kombinován s akcelerometrem je fakt, že zatímco gyroskop měří úhlovou rychlost, tak akcelerometr měří zrychlení, a tím se výborně doplňují.

Akcelerometr určuje směr, kterým se zařízení pohybuje pouze po dvou osách, rozpoznání pohybu i ve třetí ose je tudíž zajištěno právě gyroskopem. Díky tomu, může být lépe určen skutečný pohyb zařízení v prostoru.[14] Pokud je zařízení v klidu gyroskop neposkytuje žádná data. Stačí však zařízením pootočit a data o směru a rychlosti otáčení jsou již gyroskopem generována.[17]

3.2 Senzory prostředí

Operační systém Android nabízí čtyři senzory, díky kterým lze pozorovat okolní prostředí. Těmito senzory lze monitorovat relativní vlhkost vzduchu, osvětlení, tlak a teplotu v okolí zařízení s operačním systémem Android. Všechny tyto senzory jsou hardwarově založené a zařízení je obsahují jen v případě, že senzory do nich byly při výrobě vložené. Kromě senzoru osvětlení, který mnoho výrobců používá ke kontrole jasu obrazovky, nejsou vždy senzory prostředí dostupné pro všechna zařízení. Na rozdíl od pohybových a polohových senzorů, které doručují multidimenzionální řadu senzorických hodnot pro každou senzorickou událost, senzory prostředí doručují jednu senzorickou hodnotu pro každou datovou událost.[4]

3.2.1 Senzor intenzity okolního osvětlení

Tento senzor, jak již bylo zmíněno výše, se řadí ke kategorii senzorů prostředí a jedná se o senzor s hardwarovým založením.

Měří úroveň osvětlení prostředí, a to v Lx.[13] Senzor osvětlení zajišťuje automatické nastavení úrovně jasu obrazovky a jeho úkolem je stanovit jaké je v daný moment osvětlení kolem zařízení s tímto senzorem. Po vygenerování hodnoty dojde buď k většímu přisvícení obrazovky, či naopak k jejímu ztmavení. Jako světelný senzor bývá nejčastěji požit fotorezistor. [14]

3.3 Senzory polohy

Systém Android nabízí několik senzorů, díky kterým lze určit polohu daného zařízení. Těmi důležitými pro tuto bakalářskou práci jsou geolokační senzor a senzor přiblížení.

3.3.1 Senzor přiblížení

Senzor přiblížení slouží k určení, jak blízko je přední část zařízení k určitému objektu. Jedná se o senzor, který je hardwarově založený. Přenosná zařízení mívají velmi často výrobcem vložený senzor přiblížení, a to z důvodu určení, kdy je toto přenosné zařízení blízko tváře jeho uživatele, například při telefonním hovoru.[10] Praktickým příkladem fungování senzoru může být situace, kdy s někým telefonicky hovoříme a přiblížíme telefon k uchu. V tento okamžik senzor tuto skutečnost zaznamená a dochází k automatickému uzamčení obrazovky telefonu. Předchází se tím například nechtěnému ukončení hovoru.[14] Tento senzor měří jednohodnotové pole, které reprezentuje vzdálenost určenou v cm od senzoru.[21]

3.3.2 Geolokační senzor

Zařízení s Androidem může svou polohu zjistit mnoha způsoby. Nejpřesnější určení, a to až na několik metrů, probíhá skrz satelity, z nichž se sestává americká síť GPS.[22] Pro vytváření aplikací zjišťujících polohu je důležité chápat, jak GPS funguje. Jedná se o satelitní systém, který provozuje americká vláda. K tomu aby telefon mohl odečítat pozici, je zapotřebí nerušený příjem alespoň ze tří satelitů. Satelitní systém měří aktuální zeměpisnou šířku, délku a nadmořskou výšku. Zeměpisná šířka udává vzdálenost na sever či jih od rovníku, kdy severní hodnota je kladná a jižní záporná a rozsah je -90 až 90. Zeměpisná vzdálenost na východ či západ od nultého poledníku, kdy východní souřadnice jsou kladné a západní jsou záporné.[22] Geolokační senzor je tedy senzor pro určení aktuální geografické polohy zařízení. Zásluhou použitím takovéhoto senzoru je zvětšení lokačních možnosti a polohových možností aplikací. Jedná se o senzor, který je hardwarově založený. [7]

3.4 Fotoaparát a mikrofon

Jelikož fototaparát ani mikrofon nelze jednoznačně podřadit pod výše rozebrané tři kategorie senzorů, byly oba senzory zařazeny v této bakalářské práci do zvláštní kapitoly, která se věnuje pouze jim.

3.4.1 Fotoaparát

V podstatě lze fotoaparát chytrého telefonu vnímat jako jeho zrak. Fotoaparát je založen na fotosenzoru. Dnes tento senzor neslouží již pouze k pořízení fotografií či videí, ale dokáže i vytvořit panorama, rozpoznat text, určit pohlaví fotografované osoby nebo poznají kulturní či turistickou památku.[17] Jedná se o hardwarově založený senzor. Přední část fotoaparátu může být namířena buď naproti přední části obrazovky zařízení nebo je přední část fotoaparátu ta samá jako přední část obrazovky zařízení. [1]

3.4.2 Mikrofon

Tak jako fotoaparát lze považovat za zrak telefonu, lze mikrofon vnímat jako jeho sluch. Dnes mohou mít telefony i více než jen jeden mikrofon, a to především k potlačení hluku na pozadí hovoru. Také může jeden mikrofon směřovat stejným směrem jako fotoaparát, a tak bude určen k užití při natáčení videa.[17] Jedná se o senzor, který je taktéž hardwarově založený.

3.5 Shrnutí

Všechny výše uvedené senzory patří mezi nejčastěji využívané senzory při tvorbě obdobných Android aplikací bezpečnostních alarmů, jako je ten, který je výsledkem této bakalářské práce, a to pro jejich charakteristické rysy. Stačí se zamyslet, pro jaké situace bude využíván bezpečnostní alarm a jaké senzory tudíž bude zapotřebí. Pokud se rozhodnu, že chci aby mi alarm zabezpečil automobil, budu potřebovat geolokační senzor v případě, že mi s automobilem někdo odjede, a samozřejmě akcelerometr abych detekoval, že se automobil vůbec rozjel. Pokud ho použiji k zabezpečení domu, budu potřebovat senzor jako je mikrofon pro detekci zvuku a kameru pro detekci pohybu. Z výše uvedených důvodů byly tudíž pro aplikaci bezpečnostního alarmu použity následující senzory: akcelerometr, geolokační senzor, fotoaparát a mikrofon. Naopak senzor gyroskop, senzor intenzity okolního osvětlení a senzor přiblížení byly proto, že mohou být využity pouze pro minimální množinu úkonů, z tvorby aplikace bezpečnostního alarmu vyřazeny. Důvodem navíc pro vyřazení byl i fakt, že v krajním případě lze použít kameru pro stejný účel, jako by byly využity senzory přiblížení a intenzity okolního osvětlení, a v případě gyroskopu, akcelerometr je dostatečným senzorem k určení pohybu zařízení.

4 Rozbor vybraných aplikací bezpečnostních alarmů

Tato kapitola bude věnována rozboru vybraných aplikací bezpečnostních alarmů, a to především z hlediska senzorů, které ke svému účelu tyto aplikace využívají a dále z hlediska výhod a nevýhod jejich funkcionalit. K rozboru bylo vybráno celkem pět aplikací: Salient Eye, Easy security alarm, Cerberus, Dont touch my phone a Globio.

4.1 Salient Eye

4.1.1 Popis aplikace

Jako první aplikace byla k rozboru zvolena aplikace s názvem Salient Eye, Motion Detector Home securityAlarm. Jedná se o aplikaci bezpečnostní kamery. To znamená, že tato aplikace pouze detekuje pohyb kamerou. Při detekování pohybu upozorní uživatele zvukovým alarmem, emailem nebo SMS zprávou.

4.1.2 Funkce

Tato aplikace využívá pouze jeden senzor, a to senzor kamery, který detekuje pohyb. Dále může upozornit uživatele na odpojení z elektrické sítě a nebo na nízký stav baterie. K této aplikaci lze stáhnout druhou aplikaci, která umožňuje dálkové ovládání této aplikace.

4.1.3 Zhodnocení aplikace

Díky své jednoduchosti je dobrou volbou k použití na detekci pohybu neboť, ačkoli umí pouze jednu věc, tak ji umí dobře. K výhodám patří dálkové ovládání, kdy lze aplikaci vypnou či zapnout z jiného mobilního telefonu a na tento telefon dostávat i upozornění z aplikace. Naopak nevýhodou jsou placené funkce, jako je volba tónu a délky alarmu a kalendář pro automatickou aktivaci(29,33 Kč–129,99 Kč za položku).

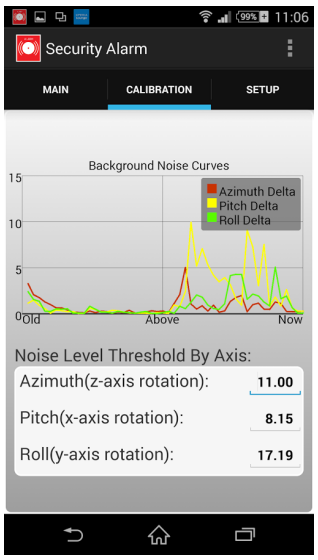
4.2 Easy security alarm

4.2.1 Popis aplikace

Další aplikací je aplikace s názvem Easy security alarm, což je aplikace, jenž detekuje pohyb zařízení pomocí akcelerometru a gyroskopu. Umí upozornit uživatele zvukovým alarmem nebo SMS zprávou.

4.2.2 Funkce

Aplikace využívá dvou pohybových senzorů - akcelerometru a gyroskopu, kterými detekuje pohyb a orientaci zařízení. Aplikace obsahuje kvalitně zpracovaný graf (Obrázek 4.1), který znázorňuje pohyb zařízení. Aplikace může běžet na pozadí a lze ji ovládat z oznamovací oblasti zařízení.



Obrázek 4.1: Graf aplikace Easy Security Alarm[3]

4.2.3 Zhodnocení aplikace

Výhodou této aplikace je zmíněný graf a možnost ovládání z oznamovací oblasti. Nevýhodou je však možnost volby pouze jedné z akcí, jak bude uživatel o o pohybu informován. Dále nastavení aplikace je dle mého názoru nevhodně popsané, kdy například telefonní číslo je označeno jako MSISDN, což pro mnoho uživatelů je označením s neznámým významem. Dále také u možnosti volby inteligentního módu zcela chybí popis této funkce. Mínusem je, že bez placené verze jsou přidány reklamy, časový limit ke spuštěný alarmu, a je deaktivovaná možnost pořízení fotografie při aktivaci alarmu.

4.3 Cerberus

4.3.1 Popis aplikace

Aplikace Cerberus slouží k nalezení ztraceného telefonu pomocí GPS senzoru. Má také možnosti ovládání řady funkcí na sledovaném telefonu.

4.3.2 Funkce

Tato aplikace využívá GPS senzoru, fotoaparátu a mikrofonu. Aplikace dokáže blokovat pokus o vypnutí zařízení nebo smazání aplikace. Z dalšího zařízení lze sledovaný telefon uzamknout, lze z něho vymazat data, zavolat a poslat SMS zprávu, nahrát video či pořídit fotografii. Díky této aplikaci lze také na druhé zařízení upozornit na stav baterie, změnu sítě, pohyb nebo pokus o smazání aplikace sledovaného zařízení.

4.3.3 Zhodnocení aplikace

Nevýhodou aplikace je především to, že je placená(131 Kč/rok), uživatelské rozhraní je nepřehledné a komplikované, a některé funkce jako například zabránění vypnutí, fungují pouze s odemknutou obrazovkou. Naopak plusem této aplikace jsou možnosti určitého ovládání telefonu z druhého zařízení jako například možnost sledovaný telefon uzamknout.

4.4 Dont touch my phone

4.4.1 Popis aplikace

Čtvrtou aplikací je aplikace s názvem Dont touch my phone. Jedná se o velmi jednoduchou aplikaci, která spustí zvukový alarm při pohybu s telefonem.

4.4.2 Funkce

Tato aplikace využívá pohybový senzor akcelerometr. Lze měnit zvuk alarmu, který se spustí při detekci pohybu.

4.4.3 Zhodnocení aplikace

Aplikace má velmi jednoduchý design. Nevýhodou však je, že ji nelze konfigurovat a obsahuje reklamy.

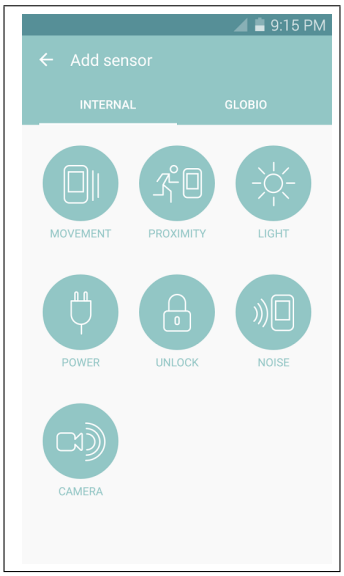
4.5 Globio

4.5.1 Popis aplikace

Poslední aplikací bezpečnostního alarmu, která bude popsána, je aplikace s názvem Globio. Aplikace má v neplacené verzi možnost detekovat zvuk, změnu světla, přiblížení k telefonu a odpojení z elektrické sítě. V placené verzi má navíc možnost detekce pohybu zařízení i kamerou. Aplikace je primárně určena k používání s hardwarem od stejné firmy.

4.5.2 Funkce

Aplikace používá tyto senzory: akcelerometr, senzor přiblížení, senzor intenzity okolního osvětlení, fotoaparát a mikrofon(Obrázek 4.2). Mezi externí hardware, který lze dokoupit od stejné firmy, patří například teploměr, magnet do dveří, infračervený detektor pohybu a tlačítko nouze. Akcemi po spuštěný alarmu jsou zvukový alarm, vibrace, pořízení audiovideo záznamu a v placené verzi poslání emailu, SMS zprávy či zavolání. Také lze od firmy dokoupit kabel, jenž dokáže poslat signál do externího zařízení.



Obrázek 4.2: Výběr senzorů aplikace Globio[5]

4.5.3 Zhodnocení aplikace

Dle mého názoru se jedná o nejkvalitněji zpracovanou aplikaci ze všech aplikací, se kterými jsem se při psaní bakalářské práce setkal. Na svojí obsáhlost

funkcí má přehledný design, ale opět mnoho funkcí je placených(60 Kč). Nevýhodou je také fakt, že aplikace nedokáže běžet na pozadí.

4.6 Celkové zhodnocení aplikací

Výše uvedené aplikace byly zvoleny z důvodu shody jejich účelu s účelem aplikace, která bude výsledkem této bakalářské práce. Všechny tyto aplikace tedy slouží právě jako bezpečnostní alarmy a ke svému účelu využívají určitých senzorů mobilních zařízení jako například senzor přiblížení, kameru, akcelerometr či gyroskop, a to konkrétně k detekci pohybu a k detekci orientace zařízení, dále GPS senzor k určení polohy zařízení, senzor intenzity okolního osvětlení k měření úrovně osvětlení okolního prostředí a mikrofon k detekci zvuku.

Jak je však patrné z jednotlivých popisů aplikací, každá aplikace využívá jiného senzoru či jiných množství senzorů. Společným mínusem těchto aplikací jsou však placené funkce, které rozšiřují možnosti funkcí aplikací, a obsažení velkého množství reklam. Po rozboru vybraných aplikací a jejich zhodnocení, lze říci, že nejkvalitnější aplikací je poslední uvedená aplikace s názvem Globio. Tato aplikace využívá celé řady senzorů, a to nejen těch základních jako je akcelerometr či senzor přiblížení, ale lze k němu dokoupit i externí hardware jako teploměr či magnet do dveří. Aplikace Globio byla tak největší inspirací při vyvíjení aplikace této bakalářské práce, a to především právě pro svou obsáhlost funkcí.

5 Rozbor aplikace

5.1 Popis aplikace

Výsledkem této bakalářské práce je aplikace, jejíž cílem je rozpoznání určité problémové situace a následné upozornění uživatele na její vznik. Jedná se tedy o Android aplikaci v roli bezpečnostního alarmu. Aplikace je navržena tak, aby mohla běžet i na pozadí a je určena pro mobilní zařízení s operačním systémem Android.

K plnění svého účelu využívá tato aplikace senzorů mobilního zařízení, na kterém je nainstalovaná. Aplikace bezpečnostního alarmu využívá více senzorů, mezi kterými je možná volba, a to z důvodu širšího použití při detekci problémové situace. Konkrétně jsou jimi: senzor kamery, mikrofonu, akcelerometr a GPS. Kamera v této aplikaci slouží k detekci pohybu a pořizování fotografií. Mikrofon detekuje zvuk, který může i nahrávat. Akcelerometr rozpoznává pohyb. GPS zaznamenává polohu mobilního zařízení. Aplikace kromě funkcí daných senzorů ovládá ještě další funkci. Tou je funkce detekce připojení do elektrické sítě.

Aplikace také obsahuje větší množství podporovaných akcí sloužících k upozornění uživatele. Tyto podporované akce můžeme rozdělit do dvou skupin. První skupina akcí jsou akce, které aplikace spustí přímo na místě určité problémové situace. Do této skupiny patří spuštění zvukového alarmu, pořízení fotografie a nahrávání zvuku. Druhá skupina akcí jsou akce, které uživatele o vzniku problémové situace informují. Uživatel si tak může nastavit, zdali chce být o dané situaci informován prostřednictvím SMS zprávy či emailem.

Vzhledem k většímu množství případů použití je zapotřebí mít možnosti přednastavení několika profilů, a to pro rychlé přepínání mezi těmito případy. Tím jsou myšleny přednastavené spouště a akce. Spouště jsou funkce, které spouští bezpečnostní alarm (např. detekce pohybu) a akce jsou funkce, které se vykonají po spuštění poplachu (např. poslání SMS zprávy uživateli).

Aplikace bezpečnostní alarmu nachází uplatnění v mnoha případech. Prostřednictvím demonstrace jednotlivých funkcí lze uvést několik případů použití:

- **Bezpečnostní kamera** - Mobilní zařízení je namířeno na hlídanou oblast. Po zaznamenání pohybu může bezpečnostní kamera danou oblast vyfotit a různými způsoby informovat uživatele o narušení oblasti.

- **GPS lokátor** - Zařízení máme například schované v automobilu. Poté co automobil opustí v aplikaci zadanou oblast, aplikace periodicky vysílá svou aktuální polohu, a to SMS zprávou či emailem.

- **Detekce pohybu zařízení**-Aplikace může například také sloužit k hlídání zavazadla ve vlaku, a to třeba ve chvíli, kdy se uživatel aplikace rozhodne ve vlaku spát a nechce, aby mu někdo během spánku zavazadlo odcizil. Uživatel tudíž nastaví práh spuštění akce výše, než je pohyb vlaku, zapne zvukový alarm a uloží mobilní zařízení do daného zavazadla. K detekci pohybu mobilního zařízení umístěného v tomto zavazadle slouží senzor akcelerometr. V momentě odcizení zavazadla se tento alarm rozezvučí.

- **Detekce připojení do elektrické sítě** - Touto funkcí je kontrola připojení do elektrické sítě. Uživatel například kontroluje, zda nedošlo k přerušení dodávky elektrického proudu na chatě, a to proto, že zde má do elektřiny připojenou ledničku. Uživatel aplikace tak zapojí mobilní zařízení do stejné elektrické zásuvky, jako je zapojená lednice a v momentě, kdy dodávka elektrického proudu na chatě bude přerušena, je o tomto přerušení skrze aplikaci bezpečnostního alarmu informován.

- **Detekce zvuku** - K detekci zvuku je využíván mikrofon. Tento senzor může být užitečný například při zabezpečení domu v uživatelově nepřítomnosti, a to proti vykradení. V okamžiku, kdy mobilní zařízení zaznamená ve svém okolí hluk, tak může spustit hned několik akcí jako například zvukový alarm či informování uživatele.

Z výše uvedených případů použití vyplývá, že tato aplikace bezpečnostního alarmu může být využita na mnohé situace a jedná se tak o užitečnou aplikaci.

Po této obecné analýze aplikace bezpečnostního alarmu budou věnovány následující kapitoly uživatelskému rozhraní, persistenci dat, spouštím akcí a samotným akcím. Konkrétně u kapitol o spouštění akcí a akcích jako takových budou popsány algoritmy použité pro získání smysluplných informací ze senzorů.

5.2 Základní komponenty aplikace

5.2.1 Aktivity

Aktivity jsou základním stavebním blokem uživatelského rozhraní většiny aplikací. Slouží jako vstupní bod pro interakci uživatele s aplikací a také pro navigaci v aplikaci. Každá obrazovka aplikace odpovídá jedné aktivitě a obsahuje funkcionalitu potřebnou pro správné zobrazení obsahu a také obsahuje kód, který provádí změny podle akcí možných na této jedné obrazovce. Historie přechodů aktivit se ukládá do zásobníku a tlačítkem *Zpět* lze navigovat zpět historií až po první aktivitu.

5.2.2 Fragmenty

Fragmenty jsou komponenty, které obsahují část uživatelského rozhraní v aktivitě. Každá aktivita může obsahovat více fragmentů, které se mění například podle velikosti obrazovky. Díky modularitě fragmentů je lze použít i ve více aktivitách.

5.2.3 Služby

Pokud je zapotřebí vykonat dlouhodobější činnost na pozadí bez potřeby uživatelského prostředí, je použita služba. Služeb je zapotřebí, neboť kód vykonávaný v aktivitách je na tyto aktivity vázaný a po přepnutí na jinou obrazovku je vykonávání toho kódu přerušeno. Ještě důležitějším důvodem pro jejich použití je fakt, že dokud kód, který běží na stejném vlákně jako překreslování obrazovek, není vykonán, není možné tuto obrazovku ovládat a aplikace přestane reagovat na vstup od uživatele. Proto jsou tvořeny služby, které nejsou vázané na zbytek chodu aplikace.

5.2.4 Manifest

Každá aplikace musí obsahovat soubor $AndroidManifest.xml$. Jedná se XML soubor obsahující nejdůležitější informace pro chod aplikace. Mezi důležité části patří název aplikace, seznam všech aktivit a služeb v aplikaci a také seznam všech oprávnění, která aplikace vyžaduje.

5.3 Uživatelské rozhraní

5.3.1 Uživatelské rozhraní obecně

Uživatelské rozhraní je tváří aplikace. Jde o vše, co uživatel u aplikace vidí a vnímá. Jedná se o jednotlivé aktivity aplikace. U této aplikace bylo cílem vytvořit uživatelského rozhraní tak, aby její užívání bylo pro uživatele co nejvíce jednoduché a lehce pochopitelné. Uživatelské rozhraní je v anglickém jazyce, jelikož se jedná o celosvětově rozšířený jazyk.

5.3.2 Profily

Aplikace podporuje velké množství kombinací dostupných funkcí, které se liší podle případu použití a jejichž možné příklady byly zmíněny výše. Tyto kombinace je vhodné nastavit pouze jednou a následně uložit pro další použití. Proto aplikace umožňuje vytvoření několika profilů, mezi kterými lze jednoduše přepínat. Pokud již není profil potřeba, může být opět jednoduše smazán.

5.4 Persistence dat

Pro správný chod aplikace je zapotřebí uchovávat data mezi jednotlivými instancemi programu. To znamená, že pokud si uživatel vytvoří profil a ukončí aplikaci, je nutné tento profil uložit pro opakované použití. Také je potřeba mít během chodu mezi jednotlivými komponenty aplikace přístupná různá data, například globální nastavení celé aplikace. Ukládat data na platformě Android lze několika způsoby:

- SQLite databáze
- Shared Preferences
- přímé uložení do souborového systému

SQLite databáze je ideální pro uchovávání dat, jejichž struktura se opakuje. V případě této aplikace se jedná o uložení profilů. Každý profil odpovídá jednomu řádku tabulky v databázi.

Shared preferneces je součástí Androd API a jedná se o rozhraní umožňující ukládat dvojce hodnot do souboru unikátního pro každou aplikaci. Jedna hodnota odpovídá klíči, podle kterého si program žádá o druhou hodnotu, jejíž obsah vyžaduje. Tato struktura je vhodná pro data, která mají být

dostupná v celé aplikaci a které se dají převést do textové podoby. Tato aplikace uvedeným způsobem ukládá globální nastavení a aktivní profil. V případě změny profilu je jeho nastavení nahráno do Shared preferences souboru a přístup do databáze je pak použit pouze při změně profilu a nebo jeho nastavení.

Dalším způsobem ukládání dat je vytvoření souboru s potřebným daty. Například je tedy možné vytvořit si XML soubor a ukládat data do něho. Přímé uložení dat v případě této aplikace je použito pouze pro uložení nahraných médií, konkrétně nahraných zvukových stop a pořízených fotografií.

5.5 Spouštění akcí

5.5.1 Detekce zvuku (Sound detection)

Spoušť detekce zvuku využívá mikrofon. Poněvadž mobilní zařízení nejsou kalibrována na konkrétní detekci hlasitosti zvuku, byla přibrána možnost jednoduché kalibrace na zvolení nulové hodnoty. Tato hodnota slouží jako relativní nula hladiny hlasitosti, avšak jednotce decibelů neodpovídá, jelikož každé zařízení vrací různé hodnoty a kalibrace by musela proběhnout na každém jednotlivém zařízení. Když uživatel provede kalibraci v hlučném prostředí, hladina tohoto zvuku bude nastavena jako nula a jednotky tak mohou být jak záporné tak kladné. Z tohoto důvodu je možné si nastavit dva prahy pro spuštění poplachu. Jeden při narušení horní hladiny zvuku a jeden pro narušení spodní hladiny zvuku. Uživatel si tak může zvolit svou klidovou hladinu zvuku a poté zadat, že si přeje spuštění poplachu poté, co hlasitost zvuku dosáhne například hodnoty plus 20 a mínus 20.

Jednotky byly co nejvíce přiblíženy hodnotě decibelů, protože však nebylo k dispozici kalibrovací zařízení, tato jednotka je pouze orientační a je na uživateli, aby se sám rozhodl, jaké prahy na základě svého testování s vlastním zařízením zvolí.

Jediná jednotka dostupná pro výpočet hlasitosti je maximum velikosti amplitudy posledního měření mikrofonem. Proto jediné, co lze změřit, je relativní hlasitost mezi dvěma měřeními. Je využito následujícího vzorce:

$$G[dB] = 20 * \log(10) \frac{A}{A_0}$$

Jedná se o vzorec pro výpočet poměru amplitud a jejich ekvivalent v dB.

Kde A je aktualní měření a A_0 je uložená referenční hodnota.[2]

5.5.2 Detekce pohybu zařízení (Device movement detection)

Spoušť detekce pohybu zařízení využívá akcelerometru. Účelem této spouště je zaznamenání jakéhokoli pohybu zařízení a je nedůležité o jaký pohyb se jedná. Z tohoto důvodu je k dispozici pouze jeden práh, který je kombinací pohybů ve všech třech osách (osa x, y, z). Senzor akcelerometru vrací hodnotu aktuálního zrychlení, ale tato hodnota není vhodná, neboť například gravitace tomuto zrychlení dává nenulovou hodnotu. Z tohoto důvodu byla jako jednotka pohybu zvolena rychlost změny absolutního zrychlení. To znamená, že zařízení, které není relativně ke svému prostředí v pohybu (např. leží na stole), nebude ukazovat nenulové hodnoty. Vzorec pro tento výpočet je následující:

$$speed = \frac{x+y+z-last\_x-last\_y-last\_z}{diffTime*10000}$$

Kde $x, y, z, last\_x, last\_y, last\_z$ jsou rychlosti na jednotlivých osách a $diffTime$ je časový rozdíl obou měření.

5.5.3 Detekce pohybu kamerou (Camera motion detection)

Spoušť detekce kamerou používá pro tuto detekci porovnávání fotografií pořízených mobilním zařízením. Rychlost sledu těchto obrázků záleží na rychlosti zařízení v pořizování fotografií. Toto pořizování je pomalé, a pokud je pohyb velmi krátký, je možné, že ho zařízení nezachytí. Jelikož aplikace může běžet na pozadí, nemohlo být zvoleno nahrání videa, a to kvůli omezení operačního systému, a tím vyřešit tento problém možností porovnávání snímků z nahrávaného videa.

Práh pro detekci pohybu je rozdíl posledních dvou porovnaných snímků. Tento rozdíl je ale uživatelsky nečitelný, jelikož se jedná o součet rozdílu pixelů, a proto byl práh nastaven konstantně při testování této funkce. Algoritmus pro porovnání jednotlivých snímku se nachází na obrázku 5.1.

```
 for (int y = 0; y < image.getHeight(); y = y + 10) {
        for (int x = 1; x < image.getWidth();
         x = x + 10) {
        int clr = image1.getPixel(x, y);
        int red = (clr & 0x00ff0000) >> 16;
        int green = (clr & 0x0000ff00) >> 8;
        int blue = clr & 0x000000ff;

        int clr2 = image2.getPixel(x, y);
        int red2 = (clr2 & 0x00ff0000) >> 16;
        int green2 = (clr2 & 0x0000ff00) >> 8;
        int blue2 = clr2 & 0x000000ff;

        int totalDiff = red - red2 + green
         - green2 + blue - blue2;
        if (Math.abs(totalDiff) > 100) {
                    totDif++;
        }
}
```

Obrázek 5.1: Algoritmus pro porovnání snímku

Algoritmus porovná rozdíl každého desátého pixelu v každé desáté řadě, a to proto, že lze předpokládat, že cíl nezabere tak malou část snímku. [18] Dále se tím velmi zrychlí výpočet rozdílu a sníží se doba, po kterou aplikace nezaznamenává pohyb. Toto rozpětí pixelů se liší podle kvality snímků a bylo kalibrováno na malé rozlišení, v kterém se fotografie pořizují.

5.5.4 Detekce připojení do elektrické sítě (Power connection)

Spoušť slouží k detekci přerušení nabíjení, což lze využít i k detekci přerušení dodávky elektřiny, poněvadž se detekuje dodávka elektřiny a nikoli zapojení kabelu do zařízení.

5.5.5 Geofencing (Geo-fence)

Spoušť slouží k detekci opuštění kruhové oblasti od zadaných souřadnic, které získává pomocí GPS senzoru. Lze zadat svoji aktuální polohu jako střed zmíněné kruhové oblasti nebo ručně nastavit zeměpisnou šířku a délku.

Pro nastavení velikosti kruhu je zadán poloměr v metrech. Pro zjištění vzdálenosti od středu oblasti je využit algoritmus na obrázku 5.2.

```
double earthRadius = 6371000; //meters
double dLat = Math.toRadians(lat2-lat1);
double dLng = Math.toRadians(lng2-lng1);
double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(Math.toRadians(lat1)) *
Math.cos(Math.toRadians(lat2)) *
Math.sin(dLng/2) * Math.sin(dLng/2);
double c = 2 * Math.atan2(Math.sqrt(a),Math.sqrt(1-a));
float dist = (float) (earthRadius * c);
```

Obrázek 5.2: Algoritmus pro zjištění vzdálenosti dvou souřadnic. *lat* a *lng* jsou zeměpisná šířka a délka

5.6 Akce

Akce jsou vykonány po spuštění poplachu.

5.6.1 Zvukový alarm (Sound alarm)

Akce začne přehrávat zvuk alarmu na uživatelsky volitelnou dobu. V případě společného použití s akcí nahrávání audio záznamu, zvukový alarm se spustí až poté, co skončí nahrávání zvukového záznamu.

5.6.2 Poslaní SMS (Send SMS)

Akce odešle SMS zprávu na vybrané telefonní číslo s vlastním textem a informací, která spoušť poplach aktiovala. Na konec SMS zprávy může být přidána GPS lokace ve formátu zeměpisné šířky a délky.

5.6.3 Poslání emailu (Send Email)

Akce odešle email na vybranou emailovou adresu s vlastním textem a informací, která spoušť poplach aktiovala. Také zde může být přidána GPS lokace ve formátu zeměpisné šířky a délky na konec emailu.

5.6.4 Nahrání audio záznamu (Record audio)

Akce začne nahrávat zvukou stopu pomocí mikrofonu a uloží jí do složky určené uživatelem. Je možné si nastavit délku nahrávání.

5.6.5 Pořízení fotografií (Take photo)

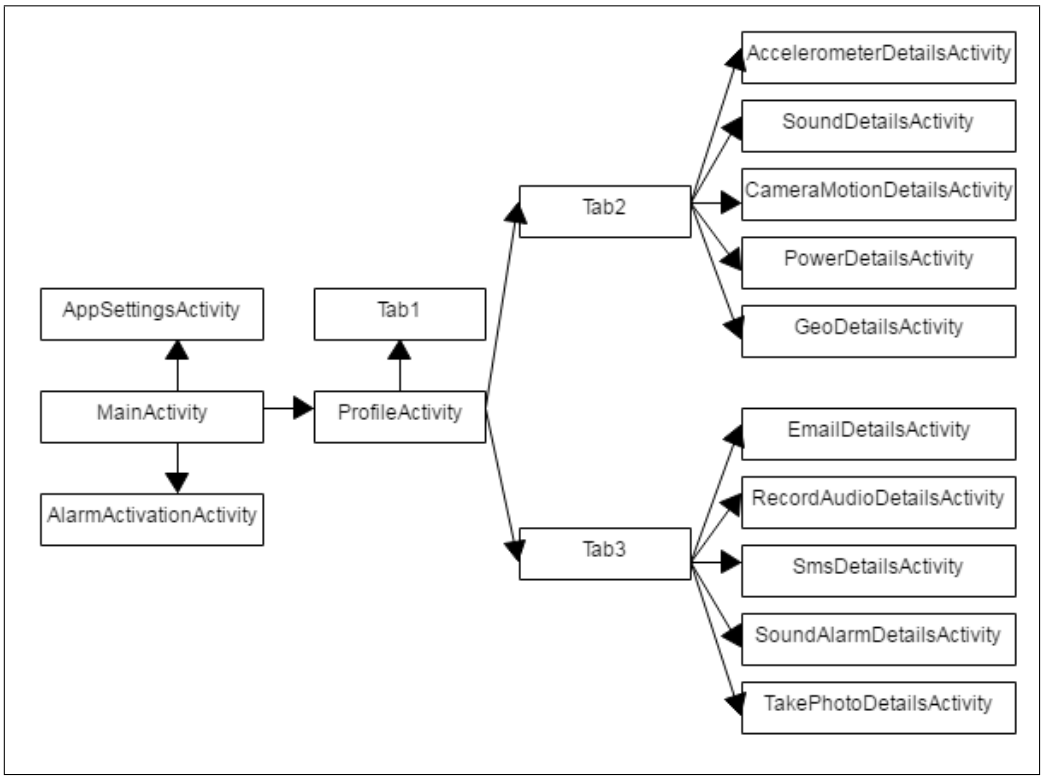
Akce pořídí fotografie primárním fotoaparátem zařízení(typicky fotoaparát na zadní straně zařízení) a uloží je do složky určené uživatelem. Lze nastavit počet fotografií a interval, ve kterém jsou fotografie foceny. Uživatel má taky možnost nastavit email, na který budou fotografie ihned odeslány.

6 Programátorská dokumentace

Tato kapitola již obsahuje konkrétní popis všech tříd tvořené aplikace bezpečnostního alarmu a jejich důležitých funkcionalit.

6.1 Aktivity

Aplikace se po spuštění otevře prostřenictvím **MainActivity** a posloupnost následujících aktivit je znázorněna na diagramu 6.1. Každá aktivita odpovídá jedné obrazovce aplikace a aktivita **ProfileActivity** využívá tří fragmentů pro vytvoření záložek, a tím místo jedné jako jediná, obsahuje tři obrazovky. Každá obrazovka může otevřít následující obrazovku podle diagramu a zpět se lze vrátit na předchozí obrazovku pomocí tlačítka zpět.



Obrázek 6.1: Diagram aktivit

Následuje popis všech aktivit a jejich důležitých metod.

6.1.1 MainActivity

Jedná se o hlavní aktivitu, která je jediným vstupním bodem do aplikace. Tato aktivita má největší množství funkcí ze všech aktivit a celkem obsahuje 22 metod. Právě z této aktivity se spouští alarm.

Obsahuje tyto funkce, které budou popsány níže:

- kontrola přístupových práv aplikace
- změna profilu
- vytvoření nového profilu
- otevření nastavení aplikace
- otevření nastavení profilu
- spuštění poplachu
- zamknutí aplikace při spuštěném poplachu

Přístupová práva

Pro udržení zabezpečení systému a uživatele, musí být aplikaci udělena přístupová práva k různým systémovým datům a funkcím. Práva, která aplikace vyžaduje jsou vypsaná v manifestu. Pro starší verze Androidu uživatel povolí tyto funkce již při instalaci aplikace na zařízení. Od verze Androidu 6 (s pojmenováním Marshmallow) se tato práva již neudělují při instalaci, ale je na aplikaci, aby si tato oprávnění vyžádala sama. Toto vyžádání oprávnění právě mimo jiné obstarává MainActivity. Důležité metody pro kontrolu práv jsou:

- **checkPermissions()** - zkontroluje zda jsou práva udělena.
- **requestPermissions()** - zažádá o práva, které nemá.

Aplikace také vyžaduje speciální povolení pro překreslování přes další aplikace. Toto povolení se od verze 6.0.5 uděluje automaticky, pokud je aplikace stažená přes Play Store. Pro povolení se otevře vlastní okno, kde uživatel funkci potvrdí a poté musí okno ručně zavřít, a to narozdíl od ostatních povolení, které lze povolit v malém okénku, které se po udělení nebo zamítnutí oprávnění samo zavře. V případě neudělení všech potřebných práv aplikace zablokuje tlačítko pro spuštění alarmu.

Změna a vytvoření profilu

Aktivita obsahuje DrawerLayout, což je menu skryté na levé straně obrazovky, které se otevře přejetím prstu z levé strany nebo tlačítkem v levém horním rohu. Ta obsahuje seznam všech uložených profilů a dále možnost vytvořit si nový profil. Nejdůležitější metody použité pro tuto funkcionalitu jsou následující:

- **drawerSetup()** - vytvoří menu se seznamem profilů.
- **setProfileToActive()** - nastaví zvolený profil jako aktivní.
- **new DatabaseHelper().addNewProfile()** - vytvoří nový profil a uloží do databáze.

Otevření nastavení aplikace

V pravém horním rohu aktivity je tlačítko, kterým lze rozbalit menu. Toto menu obsahuje položku "Settings", což otevře novou aktivitu AppSettingsActivity s nastavením.

Otevření nastavení profilu

Nastavení profilu lze otevřít stisknutím tlačítka a také je uživatel na nastavení přesměrován po vytvoření nového profilu. Metoda použité pro tuto funkcionalitu je následující:

- **launchProfileActivity()** - spustí aktivitu s nastavením profilu.

Spuštění alarmu

Po stisknutí tlačítka spuštění alarmu se aplikace zamkne a aktivuje se alarm pomocí následující metody:

- **startAlarm()** - spustí AlarmService, a tím aktivuje alarm.

Zamknutí aplikace

Pokud je alarm aktivní, nelze navigovat celou aplikací a jediná obrazovka dostupná uživateli je **AlarmActivationActivity**. Pokud uživatel spustí aplikaci, a tím i **MainActivity**, která je vstupním bodem, tak se v metodě **onResume()** zkontroluje zda není aktivní alarm. Pokud je alarm aktivní, spustí se **AlarmActivationActivity**.

6.1.2 AlarmActivationActivity

Tato aktivita slouží k ukončení aktivovaného alarmu a případě, že se alarm spustí až po uživatelem nastaveném zpoždění, aktivita zobrazuje odpočet tohoto zpoždění. Obsahuje tlačítko, které po zadání hesla deaktivuje alarm. Důležitou metodou je:

- **stopAlarm()** - zastaví alarm nebo upozorní uživatele na špatně zadané heslo.

6.1.3 AppSettingsActivity

Nastavení aplikace obsahuje možnost nastavit PIN pro deaktivaci alarmu a nastavení adresáře pro ukládání fotografií a zvukového záznamu.

PIN

Číselný kód je uložený v SharedPreferences souboru a není omezen velikostí. Pokud si uživatel nepřeje použít zabezpečení deaktivace alarmu, může pole pro zadaní PINu nechat prázdné. Uložení nového PINu je provedeno pomocí následující metody:

- **savePin()** - uloží PIN.

Nastavení adresáře

Toto nastavení je společné pro všechna média nahraná během spuštěného alarmu. Pro výběr adresáře a možnosti vytvoření nové složky pro ukládání médií byla použita malá knihovna s názvem *Android DirectoryChooser*.[16] Použití knihovny je ukázáno v kódu na obrázku 6.2

```
final Intent chooserIntent =
new Intent(this, DirectoryChooserActivity.class);
final DirectoryChooserConfig config =
DirectoryChooserConfig.builder()
        .allowReadOnlyDirectory(true)
        .allowNewDirectoryNameModification(true)
         .build();
chooserIntent
.putExtra(DirectoryChooserActivity.EXTRA_CONFIG,
 config);
startActivityForResult(chooserIntent,
 REQUEST_DIRECTORY);
```

Obrázek 6.2: Použití knihovny Android DirectoryChooser

Důležité metody pro nastavení adresáře:

- **onActivityResult()** - metoda zaznamená návrat do aplikace po zvolení adresáře .
- **handleDirectoryChoice()** - uloží adresář do **SharedPreferences** souboru.

6.1.4 ProfileActivity

Tato aktivita obsahuje veškerá nastavení profilů. Jelikož by se toto nastavení nevešlo pouze na jednu obrazovku, byly vytvořeny tři záložky, které rozdělují nastavení na tři důležité části:

- **Tab1** - obecné nastavení profilu
- **Tab2** - nastavení spouští alarmu
- **Tab3** - nastavení akcí po aktivaci alarmu

Každá tato záložka odpovídá jednomu fragmentu a budou popsány v další sekci. Vytvoření záložek je ukázáno v kódu na obrázku 6.3.

```
mSectionsPagerAdapter =
new SectionsPagerAdapter(getSupportFragmentManager());
mViewPager = (ViewPager) findViewById(R.id.container);
mViewPager.setAdapter(mSectionsPagerAdapter);
TabLayout tabLayout=(TabLayout)findViewById(R.id.tabs);
tabLayout.setupWithViewPager(mViewPager);
```

Obrázek 6.3: Vytvoření záložek

Třída *SectionsPagerAdapter* se stará o správné zobrazení záložek, a to pomocí následujících metod:

- **getItem()** - vytvoří jednotlivé záložky spuštěním fragmentů.
- **getPageTitle()** - nastaví názvy záložek v seznamu záložek.

6.1.5 Tab1 - obecné nastavení

Tab1 je fragment, který je zobrazen jako první záložka v ProfileActivity. Obsahuje obecné nastavení profilu, konkrétně následující funkce:

- změna jména profilu
- nastavení zpoždění aktivace
- nastavení zpoždění reaktivace alarmu
- možnost smazání profilu

Změna jména profilu

Jméno je po vytvoření profilu nastaveno na "new profile" a uživatel je ihned po vytvoření profilu přesměrován na tuto funkci. Jméno musí být unikátní, aby se předešlo neúmyslné záměně profilů. Funkce využívá těchto metod:

- **hasProfilesName()** - zkontroluje zda profil s tímto jménem již existuje.
- **saveName()** - uloží nové jméno.

Nastavení zpoždění aktivace

Možnost zpoždění aktivace existuje proto, aby se předešlo spuštění poplachu uživatelem ihned po aktivaci. Zpoždění umožní uživateli, dříve než se alarm aktivuje, včas opustit hlídanou oblast.

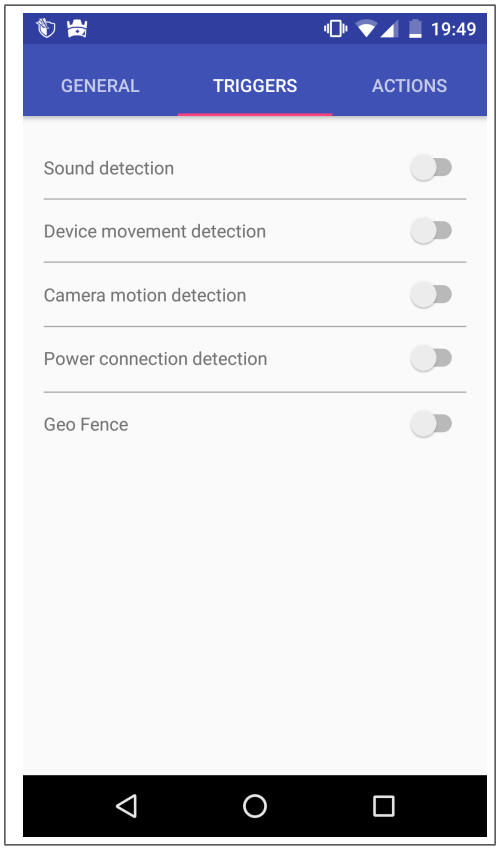
Smazání profilu

Uživatel může, pokud ho již není potřeba, profil smazat. Nemohou však být smazány všechny profily, a proto tato funkce není možná, pokud existuje pouze jeden profil. Smazání profilu používá tyto metody:

- **deleteRequest()** - zkontroluje zda se nejedná o jediný profil a spustí dialogové okno, žádající uživatele o potvrzení smazání profilu.
- **deleteProfile()** - smaže aktivní profil a ukončí **ProfileActivity**. Tím se uživatel navrátí na **MainActivity**, kde je nastaven předchozí profil za aktivní.

6.1.6 Tab2 - nastavení spouští

Tento fragment je druhou záložkou v ProfileActivity a umožňuje nastavení spouští alarmu. Obsahuje seznam všech spouští, jako je ukázáno na obrázku 6.4.



Obrázek 6.4: Screenshot aplikace na Tab2

Každá položka seznamu má dvě funkce. Kliknutím na název položky seznamu je spuštěna nová aktivita obsahující detailní nastavení zvolené spouště. Na pravé straně seznamu se nachází Switch. Jedná se o přepínatelné tlačítko, kterým lze rychle jednotlivé položky aktivovat nebo deaktivovat. Pro provedení zmíněných akcí slouží metody:

- **setOnClickListener()** - po kliknutí na textové pole s názvem položky seznamu spustí odpovídající aktivitu s detailním nastavením spouště.
- **setOnCheckedChangeListener()** - po přepnutí tlačítka zjistí jeho polohu (vypnuto/zapnuo) a uloží uživatelem zvolenou možnost.

6.1.7 Tab3 - nastevení akcí

Poslední fragment je zároveň poslední záložkou v ProfileActivity a nese podobnou funkcionalitu jako **Tab2**. Na rozdíl od umožnění nastavení spouští akcí, umožňuje nastavení zmíněných akcí. Metody použité pro nastavení akcí jsou stejné jako u **Tab2**, a tak nebudou znovu popsány.

6.1.8 Společné funkce pro aktivity s detailním nastavením

Všechny zbývající obrazovky slouží pro detailní nastavení spouští a akcí. Níže budou vypsány jejich společné důležité metody a poté u každé jejich unikátní funkce. Mezi společné funkce patří:

- editace pole
- uložení změn
- obnova prvků uživatelského rozhraní

Editace pole

Každé pole, které lze vyplnit, potřebuje klávesnici. Ta se automaticky otevře po kliknutí na editovatelné pole. Druh klávesnice se liší podle typu pole (Obrázek 6.5).

Pokud uživatel potvrdí změnu pole, musí být klávesnice zavřena pomocí kódu na obrázku 6.6.

(a) Klávesnice pro vložení telefonního čísla.

(b) Klávesnice pro vložení normálního textu.

Obrázek 6.5: Druhy klávesnic

```
public void dismissKeyboard(Activity activity) {
InputMethodManager imm = (InputMethodManager) activity
.getSystemService(Context.INPUT_METHOD_SERVICE);
if (null != activity.getCurrentFocus())
imm.hideSoftInputFromWindow(activity.getCurrentFocus()
.getApplicationWindowToken(), 0);
}
```

Obrázek 6.6: Kód pro zavření klávesnice

Uložení změn

Po editaci jakékoli možnosti se změna musí uložit do databáze. K tomu slouží následující metoda:

- **saveChange()** - uloží změny aktuálního profilu do databáze.

Obnova prvků uživatelské rozhraní

Hodnoty, které jsou v aktivitách zobrazené, jsou nastaveny při spouštění aktivity, a pokud dojde k jejich změně při spuštěné aktivitě, tato změna se neprojeví. Proto je zapotřebí periodicky kontrolovat, zda se hodnoty nezměnily. K této kontrole musí být vytvořeno nové vlákno, jelikož hlavní vlákno kontroluje uživatelský vstup, a tudíž nemůže zároveň vykonávat dlouho trvající kód. Kód pro tuto kontrolu lze vidět na obrázku 6.7.

```
runner = new Thread(){
  public void run(){
  while (runner != null){
    try{
      Thread.sleep(100);
       }catch (InterruptedException e) { };
         textView.post(new Runnable() {
         @Override
         public void run() {
           TextView textView =
           (TextView) findViewById(R.id.textViewID);
           if (textView != null) {
             textView.setText(newValue);
           }
         }
      });
    }
  }
};
runner.start();
```

Obrázek 6.7: Obnova prvku uživatelského rozhraní

6.1.9 SoundDetailsActivity

Aktivita obsahuje nastavení spouště detekce zvuku. Lze nastavit následující vlastnosti:

- referenční hladina zvuku - po kliknutí tlačítka je aktuální hladina zvuku uložena jako reference, podle které se počítá hlasitost v decibelech.
- horní práh alarmu - hranice, kterou aktuální hlasitost nesmí překročit.
- spodní práh alarmu - hranice, pod kterou aktuální hlasitost nesmí klesnout.

Aktivita také zobrazuje aktuální hlasitost zvuku kolem zařízení. Proto musí být spuštěn mikrofon a periodicky zjištěna hlasitost. K tomu slouží následující metody:

- **startRecorder()** - spustí nahrávání zvuku.

- **stopRecorder()** -zastaví nahrávání zvuku.
- **soundDb()** - volá se při obnově uživatelského prostředí (Obrázek 6.6) a vrátí aktuální hlasitost v decibelech.

6.1.10 AccelerometerDetailsActivity

Aktivita obsahuje nastavení spouště detekce pohybu zařízení. Jediná nastavitelná položka je nastavení prahu pohybu, který nesmí být překročen.

Aktivita zobrazuje aktuální rychlost pohybu zařízení a k tomuto zobrazení používá tyto metody:

- **registerListener()** - zažádá o posílání změn senzoru.
- **onSensorChanged()** - je zavolána při změně detekované senzorem.
- **unregisterListener()** -přestane zasílat změny senzoru.

6.1.11 GeoDetailsActivity

Aktivita obsahuje nastavení spouště detekce opuštění lokace.Uživatel má možnost nastavit geolokační souřadnice a poloměr v metrech. Tyto údaje dávají dohromady kruh, který nesmí zařízení opustit. Také lze nastavit souřadnice středu tlačítkem "Use current loaction", které nastaví aktuální polohu zařízení jako střed zmíněného kruhu. K tomu využívá následující metodu:

- **setCurrLocation()** - zjistí aktuální polohu a nastaví její souřadnice jako střed.

6.1.12 Ostatní aktivity

Zbylé spouště nemají k dispozici žádné nastavení a jejich aktivity zobrazují pouze krátký popis jejich funkce. Jedná se o:

- **PowerDetailsActivity** - detekce připojení do elektrické sítě
- **CameraMotionDetailsActivity** - detekce pohybu kamerou

6.2 Služby

Aplikace obsahuje dvě služby, ve kterých se nachází nejdůležitejší funkcionalita aplikace, a to vyhodnocování spouští a následné provedení všech akcí na základě nastavení.

6.2.1 Action Service

Služba se spustí ihned po tom, co uživatel spustí alarm. Životní cyklus služby je následující:

1. spuštění služby na popředí
2. inicializace všech potřebných funkcí
3. vyčkání na vypršení času pro zpoždění spuštění
4. kontrola spouští
5. pokud narazí na aktivovanou spoušť tak pokračovat, jinak vrátit na bod 4
6. provedení akcí
7. pokud je nastavená reaktivace vrátit na bod 3, jinak pokračovat
8. ukončení služby

Spuštění služby na popředí

Aby služba nebyla ukončena spolu s ukončením aplikace nebo nebyla ukončena operačním systémem, musí být spuštěna na popředí. To znamená, že po celou dobu života služby musí být viditelná notifikace, která na tento fakt uživatele upozorňuje. Toho je docíleno kódem na obrázku 6.8.

```
Notification notification = new Notification
        .Builder(getBaseContext())
        .setContentTitle("Alarm active")
        .setContentIntent(pendingIntent)
        .build();
startForeground(71, notification);
```

Obrázek 6.8: Spuštění služby na popředí

Inicializace funkcí

Služba musí začít nejdříve získávat data ze senzorů, aby mohla pokračovat ve své funkci. Pokud jsou aktivní odpovídající spouště, spustí se tyto funkce:

- získávání polohy zařízení

- získávání hlasitosti zvuku
- získávání GPS polohy
- spuštění **CameraService** pro získání detekce pohybu kamerou

Získání polohy zařízení a hlasitosti je stejné jako u již popsaných aktivit používajících tuto funkcionalitu. CameraService bude popsána v další sekci. Pro získáni polohy je použito následujících metod:

- **startGeoFence()** - připojí se k GoogleAPI a zažádá periodické získávání polohy.
- **onConnected()** - po úspěšném připojení registruje poslouchání nové polohy pomocí kódu na obrázku 6.9.
- **onLocationChanged()** - periodicky získá novou polohu a lokálně uloží pro zpracování spouští.

```
mLocationClient = new GoogleApiClient
.Builder(ActionService.this)
.addApi(LocationServices.API)
.addConnectionCallbacks(ActionService.this)
.addOnConnectionFailedListener(ActionService.this)
.build();
mLocationRequest = new LocationRequest();
mLocationRequest.setInterval(5000);
mLocationRequest.setPriority(LocationRequest
.PRIORITY_HIGH_ACCURACY);
mLocationRequest.setFastestInterval(5000);

...

LocationServices.FusedLocationApi
.requestLocationUpdates(mLocationClient,
 mLocationRequest, this);
```

Obrázek 6.9: Získávání polohy

Zpoždění spuštění alarmu

Kontrola probíhá v cyklu, kdy se postupně kontrolují všechny aktivní spouště. Ke kontrole se používají následující metody:

- **checkSound()** - získá aktuální hlasitost a porovná ji spodním a horním prahem hlasitosti.
- **checkGps()** - zjistí vzdálenost poslední získané polohy od centra určené oblasti a určí zda je mimo tuto oblast.
- **checkAccelerometer()** - určí zda pohyb překročil daný práh.
- **checkCameraMotion()** - metoda porovná velikost pohybu s nastaveným prahem, kdy tento pohyb je získán pomocí**CameraService**
- **checkPower()** - získá aktuální stav připojení do elektrické sítě pomocí kódu na obrázku 6.10.

```
IntentFilter ifilter = new IntentFilter(Intent
        .ACTION_BATTERY_CHANGED);
Intent batteryStatus = this.getApplicationContext()
        .registerReceiver(null, ifilter);
int status = batteryStatus
        .getIntExtra(BatteryManager.EXTRA_STATUS, -1);
boolean bCharging = status == BatteryManager
        .BATTERY_STATUS_CHARGING ||
status == BatteryManager.BATTERY_STATUS_FULL;
return !bCharging;
```

Obrázek 6.10: Získávání stavu připojení do elektrické sítě

Provedení akcí

Poplach je spuštěn voláním metody startAlarm(), která provede všechny aktivní akce. Používají se k tomu následující metody:

- **takePictureAction()** - předá **CameraService** informaci, že má pořídit a uložit fotografie.
- **sendEmail()** - pošle email na zadanou adresu. Využívá k tomu třídu **GMailSender** popsanou v pozdější sekci.

- **sendSms()** -pošle SMS pomocí kódu na obrázku 6.11.
- **recordAudio()** - nahraje a uloží audio záznam pomocí kódu na obrázku 6.12.
- **startSoundAlarm()** - pustí zvukový alarm.

```
SmsManager smsManager = SmsManager.getDefault();
smsManager
    .sendTextMessage(phoneNo, null, text, null, null);
```

Obrázek 6.11: Poslání SMS zprávy

```
recorder = new MediaRecorder();
recorder.setAudioSource(MediaRecorder.AudioSource.MIC);
recorder
    .setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
recorder
    .setAudioEncoder(MediaRecorder.AudioEncoder.AAC);
recorder.setOutputFile(audiofile.getAbsolutePath());
recorder.prepare();
recorder.start();
...
recorder.stop();
recorder.release();
```

Obrázek 6.12: Spuštění a ukončení nahrávání audio záznamu

Ukončení služby

V případě, že není nastavena reaktivace alarmu, služba řádně ukončí všechny funkce pro získávání dat a následně se ukončí voláním těchto metod:

- **stopForeground(true)** - ukončí běh služby na popředí.
- **stopSelf()** - ukončí službu.

6.2.2 CameraService

Práce s fotoaparátem na Android zařízeních není jednoduchá. Z bezpečnostních důvodů musí být nejdříve vytvořen náhled fotoaparátu, který musí být zobrazen uživateli. To z toho důvodu, aby bylo na první pohled vidět, že je fotoaparát aktivní a může fotit nebo nahrávat video, a tím zamezit aplikacím v používaní fotoaparátu bez jejich vědomí.

Aby aplikace mohla fungovat na pozadí, toto bezpečnostní opatření muselo být obejito a byla vytvořena služba, která tuto funkci obsluhuje.

Pořízení fotografie na pozadí bylo docíleno následujícím postupem:

1. vytvoření SurfaceView
2. spuštění náhledu na SurfaceView
3. pořízení fotografie

View je základním stavebním blokem zobrazení. Stará se o vykreslování prvků na obrazovku zařízení. SurfaceView je speciální typ View, který používá vlastní vlákno pro vykreslování a může být spuštěn přes jinou aplikaci a zároveň ji nijak neovlivňovat. Lze měnit jeho velikost i polohu. Tyto vlastnosti umožní spuštění náhledu fotoaparátu, i když je aplikace na pozadí. Velikost je nastavena na jeden pixel a náhled je umístěn neviditelný do levého horního rohu(obrázek 6.13). Poté může být pořízena fotografie a proces opakován podle potřeby.

```
windowManager =
(WindowManager) getSystemService(WINDOW_SERVICE);
params = new WindowManager.LayoutParams(
WindowManager.LayoutParams.WRAP_CONTENT,
WindowManager.LayoutParams.WRAP_CONTENT,
WindowManager.LayoutParams.TYPE_PHONE,
WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
PixelFormat.TRANSLUCENT);
params.gravity = Gravity.TOP | Gravity.LEFT;
params.width = 1;
params.height = 1;
params.x = 0;
params.y = 0;
surfaceView = new SurfaceView(getApplicationContext());
windowManager.addView(sv, params);
SurfaceHolder sHolder;
sHolder = surfaceView.getHolder();
sHolder.addCallback(this);
```

Obrázek 6.13: Vytvoření SurfaceView

Pořizování fotografií

Před každou fotografií služba rozhodne, o jaký typ fotografie se jedná. Jsou dvě následující možnosti:

1. fotografie pro detekci pohybu
2. fotografie určená k uložení na disk a poslání na email po spuštění alarmu

Detekce pohybu

Pokud má služba detekovat pohyb, kvalita pořizovaných fotografií je nastavena na velmi nízkou, aby se zvětšil počet snímků za sekundu, které zvládne služba zpracovat. Pořízenou fotografii vždy lokálně uloží a porovná s předchozím uloženým snímkem. Hodnotu rozdílu snímku uloží pro použití v ActionService a pokračuje v pořizování snímků.

Fotografie pro uložení a odeslání na email

Snímky, které se mají následně uložit, jsou foceny v nejvyšší možné kvalitě, poté jsou zkomprimovány do obrazového formátu JPEG a uloženy do adresáře určeného uživatelem. Následně je tato fotografie odeslána na zvolený email v případě, že je tato možnost aktivována.

6.3 Ostatní třídy

Poslední sekce této kapitoly popisuje třídy, které nezapadají do předchozích kategorií.

6.3.1 DatabaseHelper

Tato třída slouží k používaní SQLite databáze. Všechny třídy, které chtějí získat nebo modifikovat data v databázi, používají právě tuto třídu. Funkce, které jsou zapotřebí pro práci s databází jsou následující:

- vytvoření databáze a tabulky
- naplnění základními profily
- přidání profilu
- přístup ke všem profilům
- přístup ke konkrétním profilům
- upravování profilů
- mazání profilů

Vytvoření databáze a tabulky

Při spuštění se aplikace pokusí o načtení databáze, a pokud žádná neexistuje, tak ji vytvoří. Databáze obsahuje jedinou tabulku, a to tabulku obsahující profily. Tabulka obsahuje 33 sloupců s nastavením. Byla zvolena jedna tabulka většího rozměru místo více tabulek 1:1, protože rozdělení by nepřineslo žádnou optimalizaci, pouze zvýšilo množství kódu potřebného k manipulaci s databází. K vytvoření tabulky slouží kód na obrázku 6.14.

```
@Override
public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " +
        TABLE_NAME + " (" +
        COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + COL_NAME + " TEXT, " +
...
                  +")");
```

Obrázek 6.14: Vytvoření tabulky databáze

Základní profily

Do tabulky jsou po vytvoření vloženy dva řádky obsahující dva základní profily, které lze ihned použít. Slouží jako předloha uživateli pro první snadné použití aplikace. Profily jsou následující:

1. Detekce zvuku
    - zpoždění aktivace alarmu 10 sekund
    - detekce hladiny hlasitosti s prahy 50 a -20
    - zvukový alarm po dobu 60 sekund
    - vše ostatní deaktivováno
2. Detekce pohybu zařízení
    - zpoždění aktivace alarmu 10 sekund
    - detekce pohybu zařízení s prahem 100
    - zvukový alarm po dobu 60 sekund
    - vše ostatní deaktivováno

Přidání profilu

Pro přidání profilu je použit kód na obrázku 6.15. Všechny možnosti jsou deaktivovány a uživatel je ihned po vytvoření profilu přesměrován do nastavení k úpravě profilu.

```
SQLiteDatabase db = this.getWritableDatabase();
ContentValues contentValues = new ContentValues();
contentValues.put(COL_NAME, name);
contentValues.put(COL_DELAY, delay);
...
db.insert(TABLE_NAME, null, contentValues);
```

Obrázek 6.15: Vložení profilu do databáze

Přístup k profilům

K získání profilů z databáze je použit kód na obrázku 6.16.

```
public Cursor getAllData() {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor result = db.rawQuery("select * from " +
        TABLE_NAME, null);
        return result;
}

public Cursor getRowByID(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor result = db
        .rawQuery("select * from " + TABLE_NAME +
        " where id = " + id, null);
        return result;
}
```

Obrázek 6.16: Získání profilů z databáze

Úprava profilu

Po každé změně v nastavení profilu, jsou tyto změny ihned uloženy do databáze. Slouží k tomu metoda:

- **updateData()** - přepíše celý řádek tabulky s novými daty.

Mazání profilů

Po potvrzení smazání profilu je zavolána tato metoda:

- **deleteData(String id)** - smaže řádek tabulky.

6.3.2 Sound

Třída slouží k získávání údajů z mikrofonu. Používá k tomu následující metody:

- **startRecorder()** - spustí mikrofon.
- **getAmplitude()** - zjistí aktuální velikost amplitudy.
- **soundDb()** - spočítá relativní hlasitost v dB oproti referenční amplitudě.
- **stopRecorder()** - zastaví mikrofon.

6.3.3 SingleShotLocationProvider

Tato třída slouží k jednorázovému získání GPS polohy zařízení. K tomu využívá kód na obrázku 6.17.

```
public static void requestSingleUpdate(
final Context context, final LocationCallback callback)
throws SecurityException{

        final LocationManager locationManager =
        (LocationManager) context
        .getSystemService(Context.LOCATION_SERVICE);
        boolean isGPSEnabled = locationManager
        .isProviderEnabled(LocationManager
                .GPS_PROVIDER);
        if (isGPSEnabled) {
                Criteria criteria = new Criteria();
                criteria.setAccuracy(Criteria
                        .ACCURACY_FINE);

locationManager.requestSingleUpdate(criteria,
new LocationListener() {
        @Override
        public void onLocationChanged(
        Location location) {
                callback.onNewLocationAvailable(
                        new GPSCoordinates(
                        location.getLatitude(),
                        location.getLongitude()));
        }

        @Override public void onStatusChanged(
        String provider, int status, Bundle extras) { }

        @Override public void onProviderEnabled(
        String provider) { }

        @Override public void onProviderDisabled(
        String provider) { }
        }, null);
        }
}
```

Obrázek 6.17: Jednorázové získání GPS souřadnic

6.3.4 GMailSender

Pomocí této třídy se dají odesílat ze zařízení emaily z Gmail účtu . Pro tento účel byl vytvořen Gmail účet s adresou **Lm26Zx8BhqHLxfrx@gmail.com**. Adresa byla vygenerována náhodně, protože kombinace všech adres vystihujících funkci emailu již byli obsazeny. Z této adresy jsou odesílány všechny emaily odeslané touto aplikací. Slouží k tomu následující metody:

- **getPasswordAuthentication()** - přihlásí se k účtu.
- **addAttachment()** - připojí soubor jako přílohu.
- **sendMail()** - odešle email.

6.3.5 JSSEProvider

Jedná se o jednoduchou třídu, která poskytne JSSS - **Java Secure Socket Extension**. JSSE je balíček obsahující funkce, které umožňují zabezpečený přístup k internetu.

6.3.6 Profile

Profile je datová struktura, která uchovává dvojice dat. Konkrétně název profilu a jeho ID odpovídající řádku v databázi. Je použita seznamem profilů v **MainActivity**.

7 Testování

Vývoj aplikace probíhal primárně na zařízení Nexus 5 a byl testován na následujících zařízeních:

Model	Android verze	Rozlišení
Nexus 5	6.0.1	1080 x 1920
Samsung Galaxy A5 2017	6.0.1	1080 x 1920
HTC One V	4.0.3	480 x 800

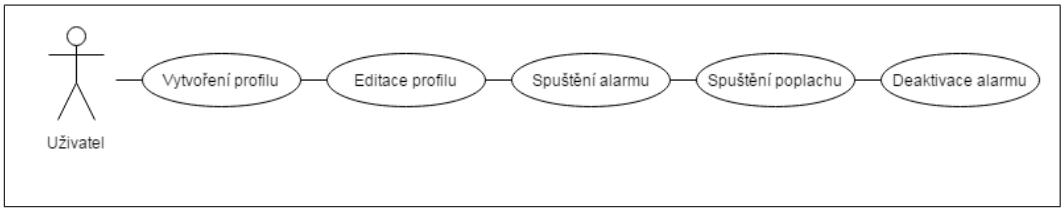
Obrázek 7.1: Tabulka testovaných zařízení

7.1 Popis testování

Při testování aplikace na každém zařízení bylo dbáno na tyto požadavky:

- správné zobrazení uživatelského rozhraní
- funkčnost všech spouští
- funkčnost všech akcí
- náročnost na výkon zařízení

Aplikace byla nainstalována na každé zařízení a bylo testováno podle scénáře znázorněného na diagramu na obrázku 7.2. Scénář byl zopakován pro všechny spouště a akce.



Obrázek 7.2: Diagram testování scénářů

Testovací scénáře

Jako testovací scénáře byly zvoleny motivační příklady popsané v kapitole 5.1.

1. **Bezpečnostní kamera**
   Testované funkce: Detekce pohybu kamerou, pořízení fotografií, odeslání fotografií na email.

   Zařízení bylo umístěno do místnosti a aktivován alarm. Poté se prošlo místností a zjišťovalo, zda zařízení pořídilo fotografie a odeslalo je na zadaný email.

2. **GPS lokátor**
   Testované funkce: Geo fence, odeslání SMS polohou, odeslání emailu s polohou, periodická reaktivace alarmu.

   Zařízení bylo umístěno do osobního automobilu a aktivován alarm. Po rozjetí automobilu a opuštění nastavené oblasti se zjišťovalo, zda zařízení periodicky zasílá svou polohu pomocí SMS a emailu.

3. **Detekce pohybu zařízení**
   Testované funkce: Detekce pohybu zařízení, zvukový poplach.

   Na zařízení byl aktivován alarm a po pohybu zařízení byl očekáván zvukový alarm.

4. **Detekce připojení do elektrické sítě**
   Testované funkce: Detekce připojení do elektrické sítě, odeslání SMS s vlastním textem.

   Zařízení bylo zapojeno do elektrické sítě a byl aktivován alarm a nastaven vlastním text. Poté byl vypojen přívod elektřiny do nabíječky a očekávala se SMS s textem: “Nejde ti proud k ledničce na chatě”.

5. **Detekce zvuku**
   Testované funkce: Detekce zvuku, nahrání audio záznamu.

   Zařízení bylo umístěno do místnosti a aktivován alarm. Poté začla konverzace v blízkosti zařízení a zjišťovalo se zda byla konverzace nahrána.

7.2 Výsledky testování

1. **Bezpečnostní kamera**
   Testování probíhalo za různých světelných podmínek a hladina prahu byla upravena pro optimální citlivost na základě výsledků testování.

2. **GPS lokátor**
   Testování proběhlo na dlouhé opuštění lokace i pohyb po hranici oblasti a nebyl zaznamenán problém s vyhlášením poplachu po splnění podmínek.

3. **Detekce pohybu zařízení**
   Byla zjišťována detekce pohybu jak v klidovém stavu, tak v pohybujícím se dopravním prostředku.Nebyly zjištěny výkyvy způsobující falešné spuštění poplachu při dobře zvoleném prahu.

4. **Detekce připojení do elektrické sítě**
   Testovala se funkcionalita pro připojení do elektrické sítě přes USB a nabíjení přímo ze sítě. Při vypojení napájení došlo spolehlivě k aktivaci poplachu.

5. **Detekce zvuku**
   Bylo testováno pro různé hladiny hluku pro klidovou hodnotu a prahů. Dosažené výsledky odpovídaly přdpokladům.

Jediná funkce, jejíž kvalita se významně lišila pro každé zařízení, byla rychlost pořízení fotografie, a poté její porovnání s předchozím snímkem pro detekci pohybu. Rozdíly jsou vidět v následující tabulce:

Model zařízení	průměrná rychlost
Nexus 5	2400ms
Samsung Galaxy A5 2017	750ms
HTC One V	1200ms

Obrázek 7.3: Testování rychlosti pořízení fotografií

8 Možná rozšíření

Díky povaze zadání práce, tedy vytvoření aplikace bezpečnostního alarmu, existuje velké množství možných rozšíření této aplikace. V úvahu přichází například přidání dalších spouští akcí, které by využívaly zbylé senzory, kterými jsou vybaveny mobilní zařízení (např. gyroskop, senzor přiblížení či senzor intenzity okolního osvětlení) nebo externích senzorů (např. teploměr či magnet do dveří jako bylo možné pozorovat u aplikace Globio) nebo i čistě softwarové spouště jako například detekce připojení k internetu, odemknutí telefonu, stav baterie atd. V případě akcí, které by následovaly po spuštění bezpečnostního alarmu, by bylo možné například přidat zamknutí či vypnutí zařízení, nahrání videa, uložení pořízených záznamů na externí úložiště nebo internet. I současná nastavení by samozřejmě mohla být později obohacována o další funkce jako například o citlivost detekce pohybu fotoaparátem.

Ačkoli, jak je patrné z výše uvedeného, je možných rozšíření celá řada, aplikace bezpečnostního alarmu vytvořená v této bakalářské práci při svém počtu spouští akcí a akcí samotných splňuje svůj účel zadaný v zadání této práce. Do budoucna je však možné, že bude o tato možná rozšíření obohacena.

9 Závěr

Cílem této bakalářské práce bylo vytvoření aplikace, která by umožnila využívat mobilní zařízení s operačním systémem Android jako bezpečnostní alarm.

Prvním úkolem bylo vybrat, jaké senzory mobilního zařízení bude tato aplikace ke svému účelu využívat. Po zvážení situací, pro které bude aplikace určena, byly zvoleny tyto senzory: akcelerometr, GPS senzor, kamera a mikrofon. Ke splnění účelu aplikace byly tyto vybrané senzory po testování zhodnoceny jako dostatečné, neboť účel splnily. Aplikace tudíž pomocí uvedených senzorů rozpozná problémovou situaci a následně na její vznik uživatele upozorní. Navíc byla přidána funkce detekce připojení do elektrické sítě.

Dalším úkolem bylo se touto aplikací, alespoň určitým způsobem, vymezit oproti vybraným obdobným aplikacím. K porovnání funkcionalit bylo vybráno pět aplikací, které ke svému účelu také využívají senzorů mobilních zařízení. Jejich společnou nevýhodou byl především fakt, že obsahují placené rozšiřující funkce a reklamy.

Vyvinutá aplikace obsahuje relativně široké spektrum využití dle výše zmíněných senzorů a podporuje velké množství kombinací dostupných funkcí, které se liší podle případů použití, které byly v této bakalářské práci popsané. Tím se vymezuje oproti většině v této práci vybraných obdobných aplikací. Vytvořená aplikace obsahuje uživatelské rozhraní, které má jednoduchý design a snadno se ovládá. Po spuštění alarmu může uživatel v aplikaci nastavit několik akcí a to spuštění zvukového alarmu, poslání SMS zprávy či emailu, a to i s GPS lokací zařízení, dále nahrání audio záznamu či pořízení fotografií. Aplikace je kompatibilní s mobilními zařízeními s Android verzemi od verze Android 4.0.3 Ice Cream Sandwich (API level 15) a výše. Důležité je také zmínit, že se jedná o základní verzi bezpečnostního alarmu, jehož funkcionality je dále možné rozšiřovat, a to jak o další spouště akcí, tak akce samotné. Cíl této bakalářské práce však lze touto verzí bezpečnostního alarmu považovat za splněný, neboť aplikace byla úspěšně implementována a splňuje požadavky kladené na ni v zadání práce.

Přehled zkratek

API	Application programming interference - rozhraní pro programování aplikací
APK	Android application package - souborový formát pro Android aplikace
C++	programovací jazyk
g	gravitační zrychlení země
GPS	global position system - mobilní navigace
IDE	Integrated Development Environment - integrované vývojové prostředí
MSISDN	Mobile Subscriber ISDN Number - telefoní číslo
NDK	Native development kit - sada vývojových nástrojů pro nativní vývoj
PIN	Personal identification number - osobní identifikační číslo
SMS	Short message service - služba krátkých textových zpráv
XML	Extensible Markup Language – Rozšiřitelný značkovací jazyk

Seznam obrázků

Literatura

[1] *Camera. Camera info* [online]. 2015. [cit. 2017/06/06]. Android developers. Dostupné z: `https://developer.android.com/guide/topics/sensors/Camera.CameraInfo.html`.

[2] *Decibel* [online]. [cit. 2017/06/19]. Dostupné z: `http://www.rapidtables.com/electric/decibel.htm`.

[3] *Easy Security Alarm* [online]. 2017. [cit. 2017/06/21]. Dostupné z: `https://play.google.com/store/apps/details?id=lv.INA.easyalarm`.

[4] *Environment sensors* [online]. 2015. [cit. 2017/06/06]. Android developers. Dostupné z: `https://developer.android.com/guide/topics/sensors/sensors_environment.html`.

[5] *Globio* [online]. 2017. [cit. 2017/06/21]. Dostupné z: `https://play.google.com/store/apps/details?id=com.alarmsystemlite.focus`.

[6] *Introduction to Android* [online]. 2015. [cit. 2017/06/01]. Android developers. Dostupné z: `https://developer.android.com/guide/index.html`.

[7] *Location and sensors APIs* [online]. 2015. [cit. 2017/06/06]. Android developers. Dostupné z: `https://developer.android.com/guide/topics/sensors/index.html`.

[8] *RMeet Android Studio* [online]. 2016. [cit. 2017/06/07]. Android developers. Dostupné z: `https://developer.android.com/studio/intro/index.html`.

[9] *Motion sensors* [online]. 2015. [cit. 2017/06/05]. Android developers. Dostupné z: `https://developer.android.com/guide/topics/sensors/sensors_motion.html#sensors-motion-linear`.

[10] *Position sensors* [online]. 2015. [cit. 2017/06/06]. Android developers. Dostupné z: `https://developer.android.com/guide/topics/sensors/sensors_position.html`.

[11] *Requesting Permissions at Run Time* [online]. 2016. [cit. 2017/06/01]. Android developers. Dostupné z: `https://developer.android.com/training/permissions/requesting.html`.

[12] *Sensor cooedinate system* [online]. 2015. [cit. 2017/06/05]. Android developers. Dostupné z: https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords.

[13] *Sensors overview* [online]. 2015. [cit. 2017/06/05]. Android developers. Dostupné z: https://developer.android.com/guide/topics/sensors/sensors_overview.html.

[14] *Senzory v mobilních telefonech od A do Z* [online]. 2017. [cit. 2017/06/05]. Beryko. Dostupné z: https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-coords.

[15] *Smartphone OS Market Share, 2016 Q3* [online]. 2016. [cit. 2017/06/05]. IDC. Dostupné z: http://www.idc.com/promo/smartphone-market-share/os.

[16] *Android DirectoryChooser* [online]. 2013. [cit. 2017/06/21]. Dostupné z: https://github.com/passy/Android-DirectoryChooser.

[17] Chroust, M. *Smartphony mají 19 smyslů* [online]. 2015. [cit. 2017/06/06]. Dostupné z: http://www.mobilmania.cz/clanky/smartphony-maji-19-smyslu-znate-je-vsechny/sc-3-a-1329584/default.asp.

[18] Elaydi, H. INTELLIGENT MOTION DETECTION AND TRACKING SYSTEM. 2010. Dostupné z: http://site.iugaza.edu.ps/helaydi/files/2010/02/ACIT_Elaydi2002.pdf.

[19] Grant, A. *Android4. průvodce programováním mobilních aplikací.* Albatros Media, 2013. ISBN SBN 978-80-251-3782-6.

[20] Újbanyai, M. *Programujeme pro Android.* Grada Publishing, 2012. ISBN 978- 80-247-3995-3.

[21] Mednieks, Z. *Programming Android, 2nd Edition.* O'Relly Media, Inc., 2012. ISBN 978- 14-493-1664-8.

[22] Wolber, D. *App Inventor.* Computer press, 2014. ISBN 978-80-251-4195-3.

Seznam příloh

- Příloha A: Instalace aplikace
- Příloha B: Uživatelská příručka
- Příloha C: Diagram databáze
- Příloha D: Struktura přiloženého CD

A Instalace aplikace

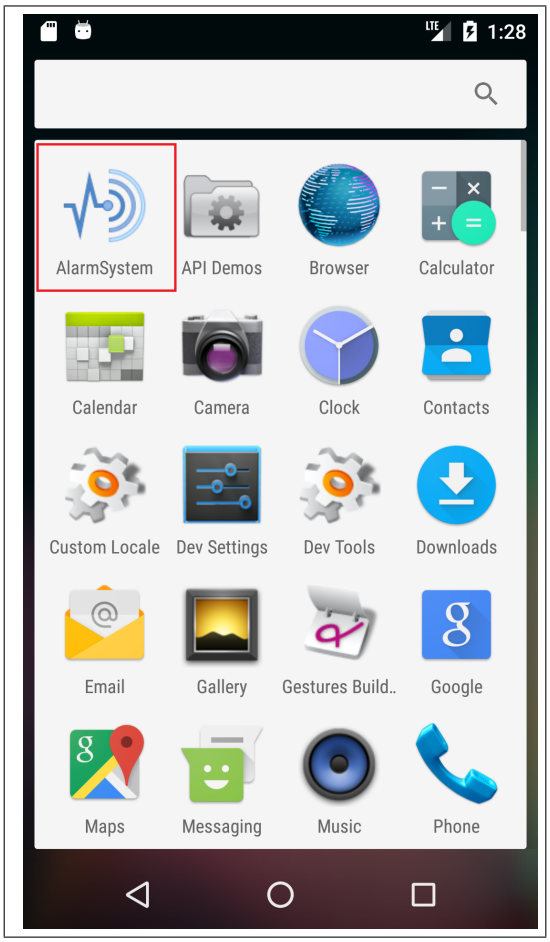
Pro instalaci aplikace do mobilního zařízení je potřeba povolit možnost instalace z neznámých zdrojů. Tato možnost se nachází v *Nastavení -> Zabezpečení -> Neznámé zdroje*.

Na přiloženém CD se nachází soubor s názvem SecurityAlarm.apk a po jeho stažení do zařízení lze jeho spuštěním aplikaci nainstalovat. Poté je aplikace připravena k použití.

B Uživatelská příručka

B.1 Spuštění Aplikace

Aplikaci lze spustit stisknutím příslušné ikony v menu zařízení(Obrázek B.1).

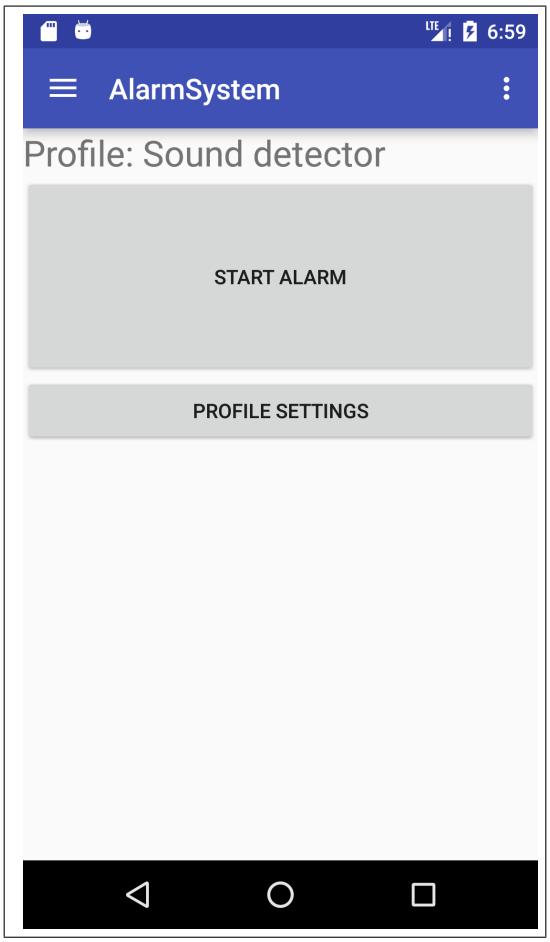


Obrázek B.1: Ikona aplikace

B.2 Ovládání aplikace

B.2.1 Hlavní obrazovka

Po otevření aplikace lze vidět hlavní stránku aplikace(Obrázek B.2).



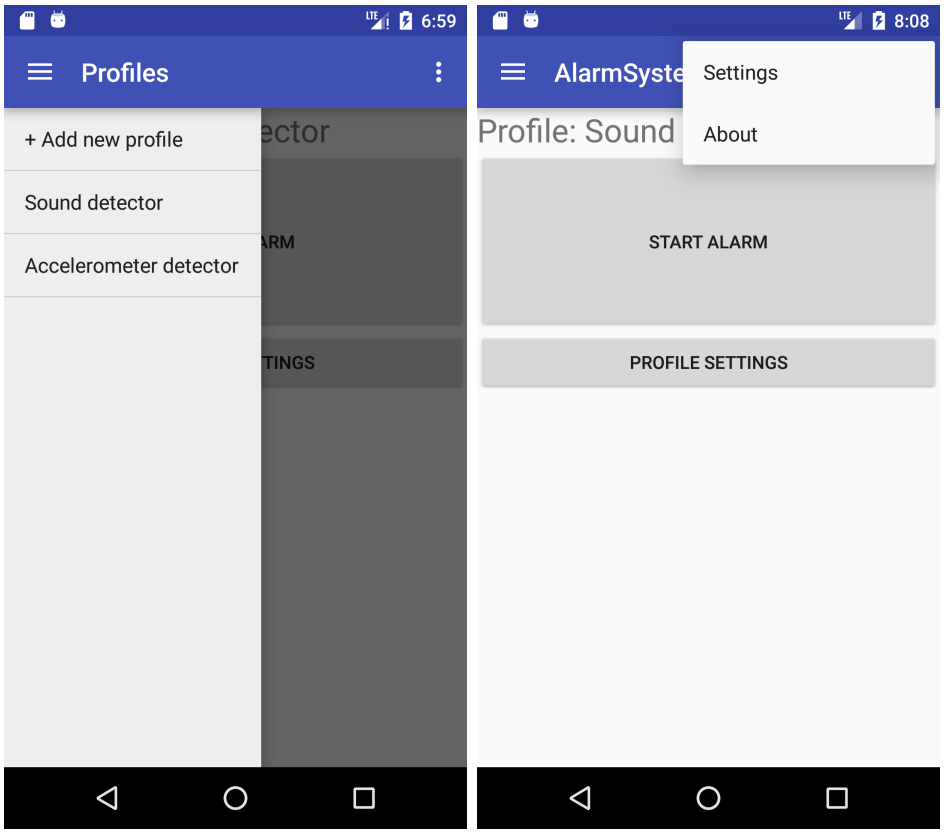
Obrázek B.2: Hlavní obrazovka

- **Start alarm**
  Stisknutí tlačítka aktivuje bezpečnostní alarm.

- **Profile settings**
  Stisknutí tlačítka otevře obrazovku s nastavením profilu.

- **Profiles**
  Tlačítko v levém horním rohu nebo přejetí prstem zleva doprava otevře výběr profilů. Po stisknutí na položku v seznamu je vybraný profil nastaven jako aktivní(Obrázek B.3.a).

- **Settings**
  Stisknutím tlačítka v pravém horním rohu otevře seznam možností. Stisknutím možnosti Settings se otevře obrazovka s nastavením aplikace.(Obrázek B.3.b)

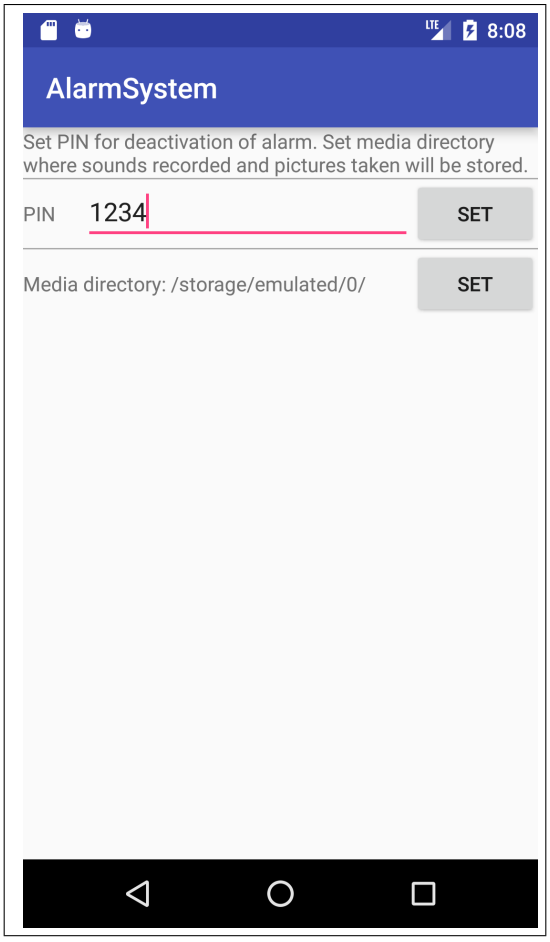


(a) Výběr profilů. (b) Hlavní menu.

Obrázek B.3: Skryté komponenty v hlavní obrazovce

B.2.2 Nastavení

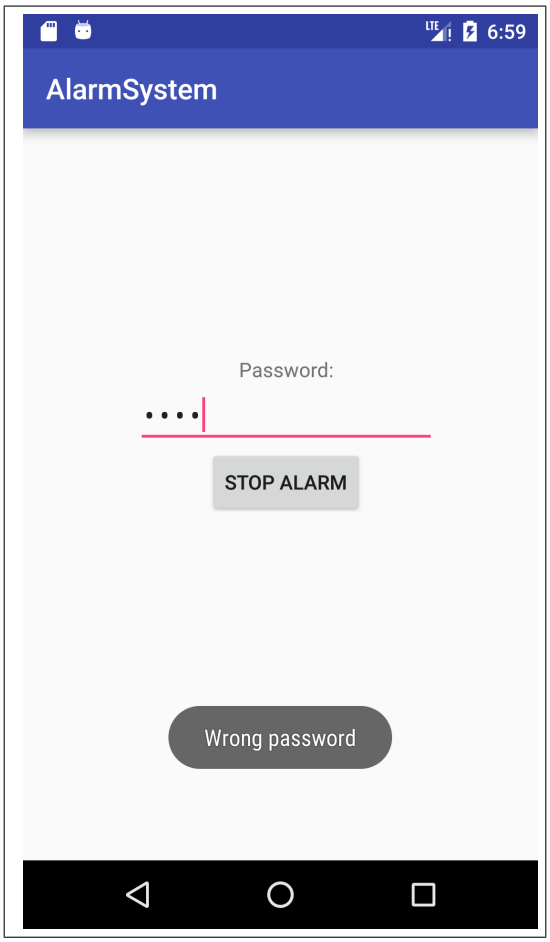
Na obrazovce s nastavením lze měnit PIN a adresář, do kterého se ukládají pořízené média.(Obrázek B.4)



Obrázek B.4: Obrazovka nastavení

B.2.3 Obrazovka aktivního alarmu

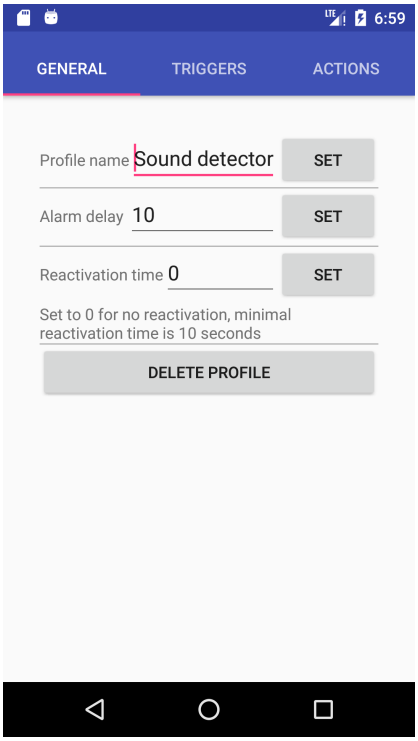
Obrazovka obsahuje textové pole pro zadání PINu pro deaktivaci alarmu a tlačítko pro potvrzení. V případě zadání špatného hesla, uživatel je o tom informován.(Obrázek B.5)



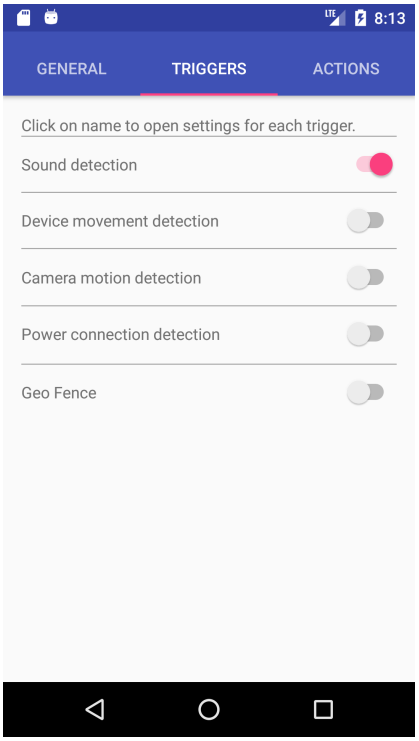
Obrázek B.5: Obrazovka aktivního alarmu

B.2.4 Nastavení profilu

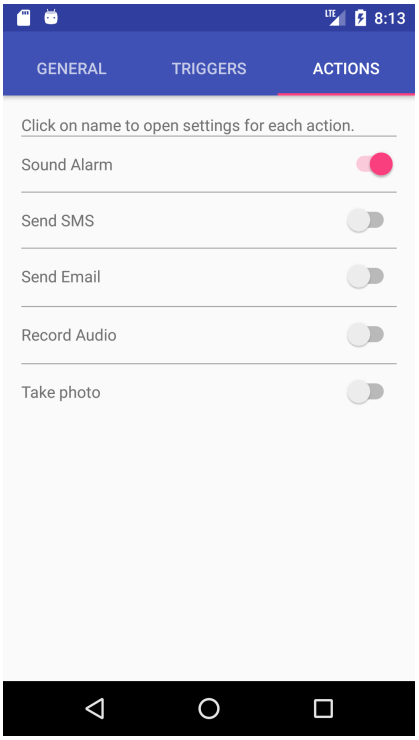
Nastavení profilu obsahuje tři záložky(Obrázek B.5).



(a) Obecné nastavení



(b) Spouště



(c) Akce

Obrázek B.6: Záložky nastavení profilu

1. **General** - Obecné nastavení (Obrázek B.6.a)
   Na obrazovce lze měnit následující:

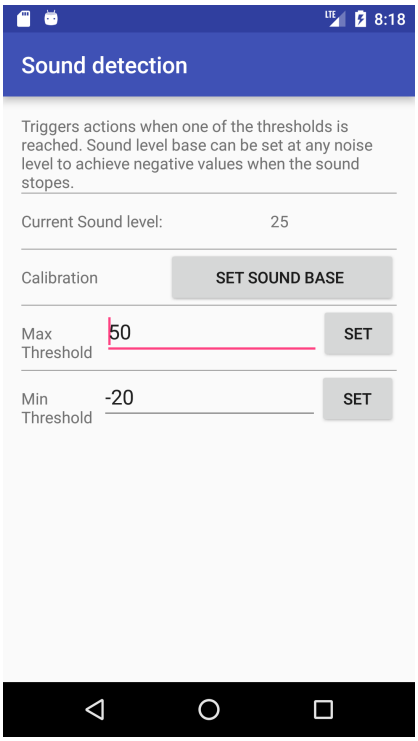
   - **Profile name** - Jméno profilu
     Jméno musí být unikátní pro každý profil.
   - **Alarm delay** - Zpoždění spuštění
     Alarm se aktivuje až po vypršení časové prodlevy.
   - **Reactivation time** - Reaktivace
     Lze nastavit po jak dlouhé době se alarm po spuštění alarmu reaktivuje.Minimální doba je 10 sekund. Nastavením hodnoty na 0 reaktivace neproběhne vůbec.
   - **Delete profile** - Smazání profilu Stisknutí tlačítka otevře dialogové okno pro potvrzení smazání profilu. Nelze smazat poslední profil.

2. **Triggers** - spouště akcí(Obrázek B.6.b)
   Obrazovka obsahuje seznam všech spouští alarmu. V pravé části lze přepínacím tlačítkem nastavit spoušť jako aktivní nebo neaktivní. Kliknutím na název spouště se otevře detailní nastavení pro každou spoušť.

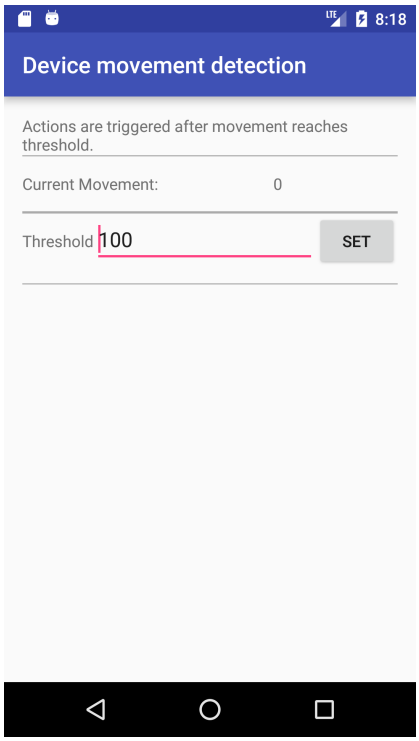
3. **Actions** - Akce(Obrázek B.6.c)
   Obrazovka obsahuje seznam všech akcí pro vykonání po spuštění alarmu. V pravé části lze přepínacím tlačítkem nastavit akci jako aktivní nebo neaktivní. Kliknutím na název akce se otevře detailní nastavení pro každou akci.

B.2.5 Detail spouští a akcí

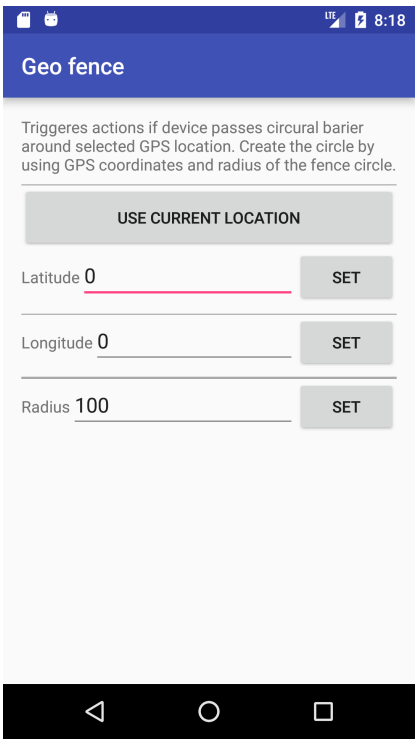
Každá nastavitelná možnost se potvrzuje tlačítkem Set. O každém úspěšném provedení uložení hodnoty je uživatel informován vyskakujícím okénkem v dolní části obrazovky.



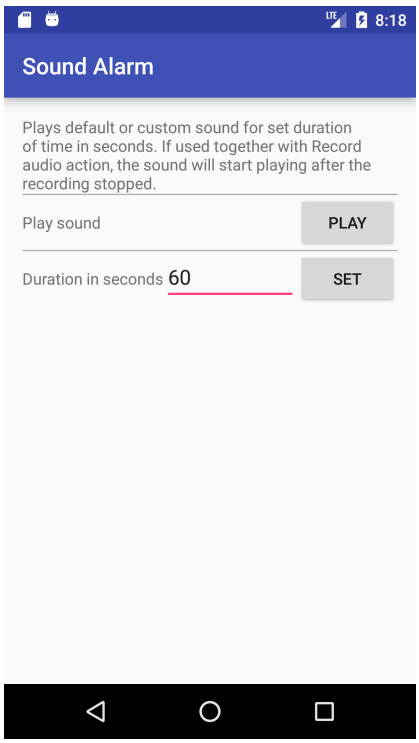
(a) Nastavení detekce zvuku



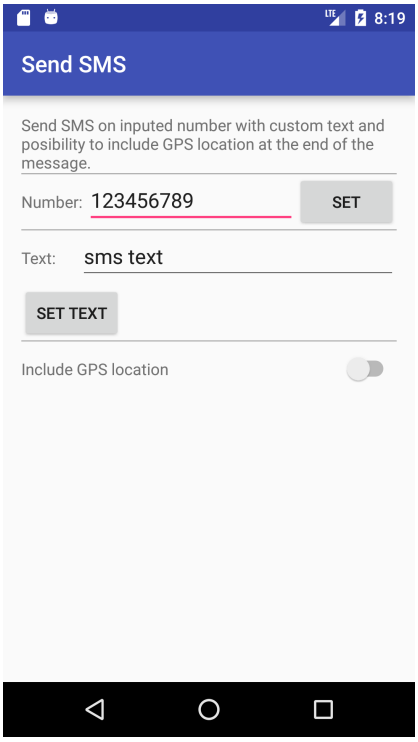
(b) Nastavení detekce pohybu



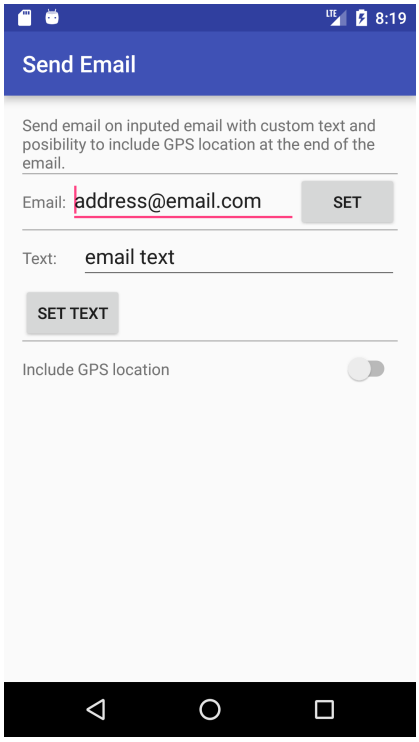
(c) Nastavení Geo-fencingu



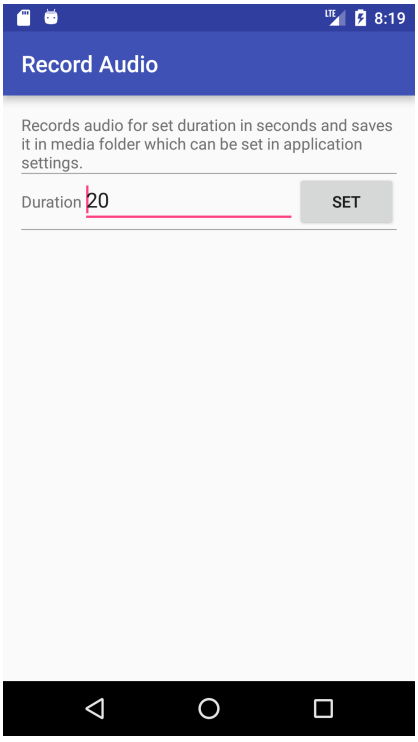
(d) Nastavení zvukového alarmu



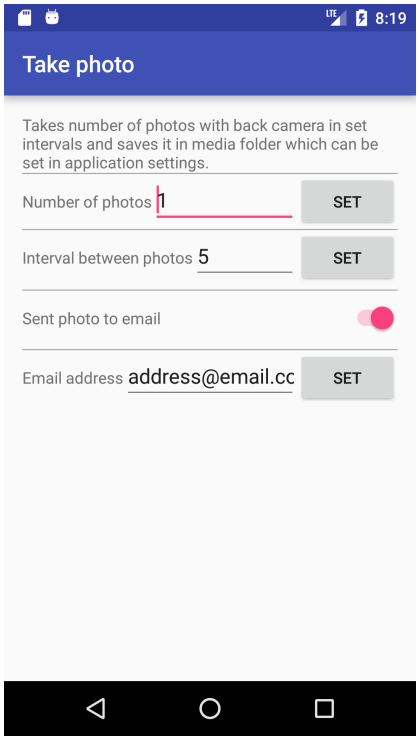
(e) Nastavení poslání SMS



(f) Nastavení poslání emailu



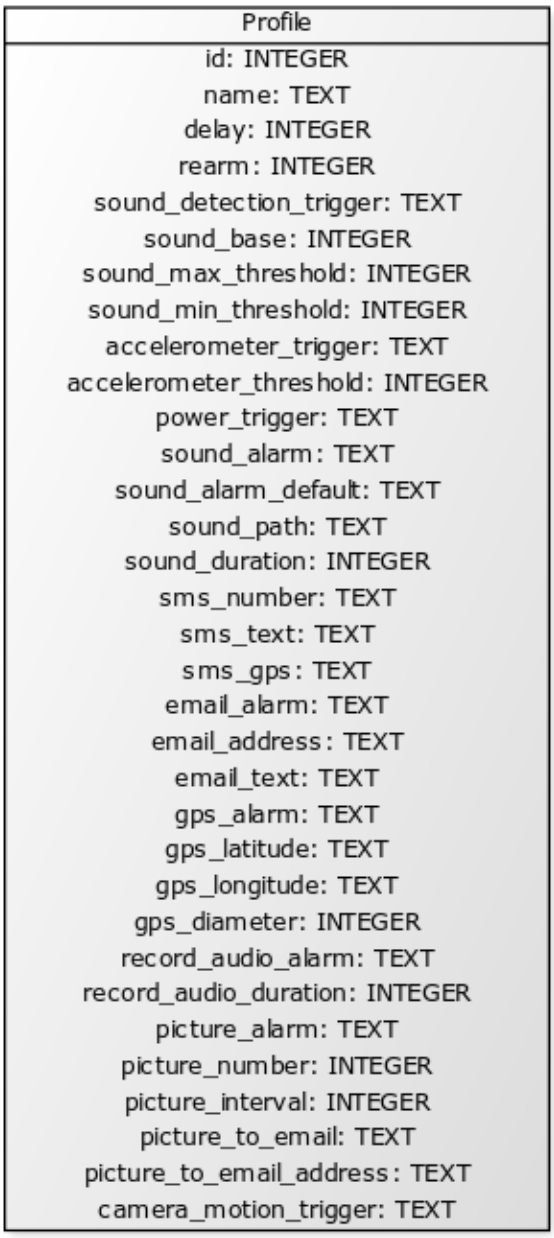
(g) Nastavení nahrání zvuku



(h) Nastavení pořízení fotky

Obrázek B.7: Detailní nastavení spouští

C Diagram databáze



Obrázek C.1: Databáze

D Struktura přiloženého CD

1. alarmSystem.apk - Instalační soubor aplikace.
2. bakalářská práce.pdf - Dokument bakalářksé práce v PDF.
3. javadoc - Adresář obsahuje dokumentaci projektu.
4. project - Adresář obsahuje zdrojové kódy aplikace.
5. tex - Adresář obsahuje zdrojové kódy dokumentu.