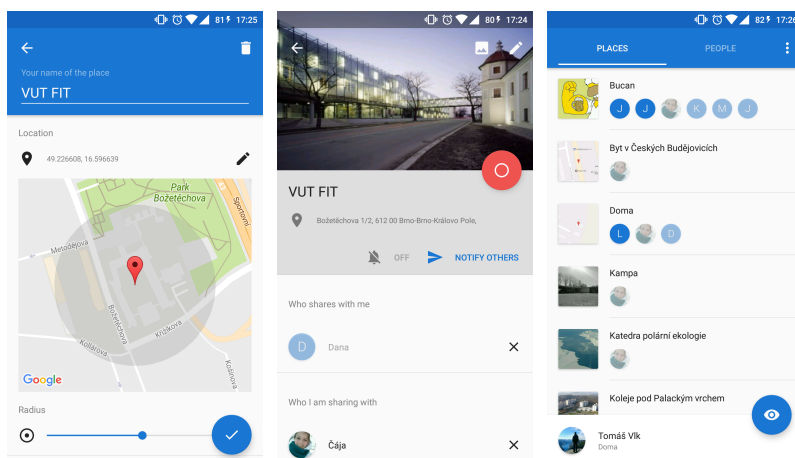


Mobilní aplikace pro sdílení informace o přítomnosti na místě

Tomáš Vlk*



Abstrakt

Cílem této práce je vytvořit uživatelsky přívětivou mobilní aplikaci, díky níž bude snazší sdílet svoji polohu a zároveň zachovat své soukromí. Toho je dosaženo sdílením pouze informace o přítomnosti na místě místo GPS souřadnic aktuální polohy. Pro snazší sdílení nabízí aplikace automatickou detekci přítomnosti na určitém místě, řešenou pomocí **GeofencingApi**. Jako cílová platforma této aplikace byl zvolen systém Android. Ke komunikaci mezi zařízeními se využívá **Firebase Cloud Messaging**. Výsledkem práce je, že s mírnými úpravami funguje **GeofencingApi** spolehlivě a lze určit přítomnost zařízení na místě s poloměrem větším než 50 metrů. Zároveň ale nedochází k aktivnímu dotazování polohy. Podařilo se tak vytvořit aplikaci, která nemusí být vůbec spuštěna (ani žádná její služba na pozadí) a přitom dokáže detekovat přítomnost zařízení na určitém místě a přijímat zprávy od ostatních zařízení. Největší předností tohoto přístupu je velmi nízká spotřeba energie oproti jiným řešením.

Klíčová slova: Android — Firebase — Poloha — Geofence — Sdílení

Přiložené materiály: [Google Play](#)

*xvlkto00@stud.fit.vutbr.cz, *Fakulta informačních technologií, Vysokého učení technického v Brně*

1. Úvod

Lidé chtějí vědět, kde se nachází jejich blízcí a známí. Ne každý ale chce svou polohu sdílet. Tato aplikace vznikla za účelem tento problém vyřešit a najít tak kompromis. Aby uživatelé této aplikace měli nad svým soukromím kontrolu, nesdílí aplikace GPS souřadnice aktuální polohy, ale pouze informaci o přítomnosti na určitém místě. Dalším stupněm ochrany soukromí je

definování osob u každého místa, kterým tato informace bude sdílena. Důležitou součástí aplikace je pak také automatická detekce přítomnosti na místě a sdílení této informace. Uživatel tak velmi zjednodušuje ovládání.

Tato práce řeší problém vytvoření automatické detekce přítomnosti na místě tak, aby byla spolehlivá a co nejméně energeticky náročná. Další klíčová věc

je zprostředkování informace o přítomnosti na místě ostatním uživatelům. K tomuto účelu musí být v celém systému zapojený server, ke kterému se klientské aplikace uživatelů budou připojovat.

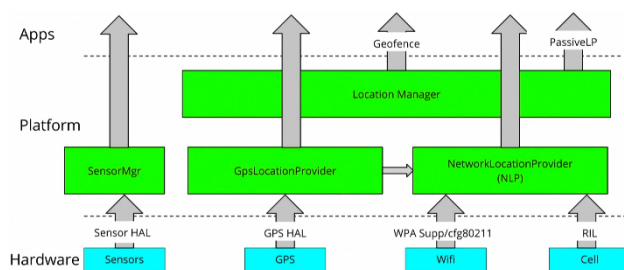
Většina existujících aplikací, nabízející sdílení polohy, funguje na principu sdílení GPS souřadnic v reálném čase. Aplikace tohoto typu často nabízejí možnost omezit přístup k poloze jen určitým lidem nebo jen po určitý čas. Příkladem mohou být aplikace *Glympse* a *Life360*. Nově pak tuto funkci nabízí i *Google Maps* nebo *Facebook Messenger*. Ne vždy je ale tento přístup ideální. Sdílet někomu GPS souřadnice své aktuální polohy je velký zásah do soukromí. Omezení sdílení polohy po určitý čas může tento problém částečně řešit, ale vyžaduje aktivní zapojení uživatele. Uživatel musí aplikaci otevřít a nastavit čas, po který se poloha bude sdílet a případně osoby, kterým tato poloha bude sdílána.

Tato aplikace řeší sdílení polohy jinak. Uživatelé mají svůj seznam míst. Každé toto místo má své GPS souřadnice a velikost (poloměr). Následně si uživatelé mezi sebou nastaví, kdo je na jednotlivých místech může vidět. Aplikace pak automaticky detekuje přítomnost na těchto místech a sdílí pouze tuto informaci, nikoliv GPS souřadnice aktuální polohy.

Uživatel si tak může být jist, že ho nikdo nemůže sledovat. Jakmile dorazí na místo, u kterého svou přítomnost někomu sdílí, ví přesně komu a může toto nastavení okamžitě ovlivnit.

2. Prostředky pro automatickou detekci přítomnosti na místě

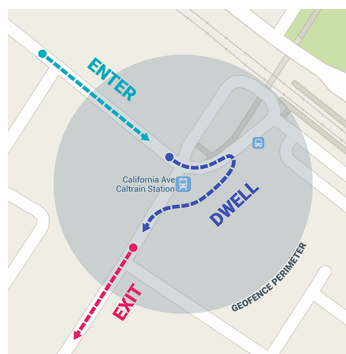
Každý mobilní telefon v současné době disponuje GPS modulem, který má přesnost přibližně několik metrů [1]. Zapnutí tohoto modulu je ale velmi energeticky náročné. V Androidu se dá určit poloha i z WiFi nebo mobilní sítě. Poloha z WiFi má přesnost zhruba 50 až 500 metrů, u mobilních sítí to je pak 500 metrů až několik kilometrů [1]. Kombinací všech zdrojů polohy lze získat nejrychleji a nejpresněji aktuální polohu. Android nabízí rozhraní *LocationManager* [2], které zastřešuje všechny zdroje polohy. Schéma tohoto API je pro lepší představu na obrázku 1. Proces získání polohy může aplikace urychlit a zefektivnit pokud se společně s tímto API využijí například informace ze senzorů, poloha získaná jinými aplikacemi nebo historie polohy. Proto existuje tzv. *FusedLocationProviderApi*, které tyto další faktory bere v potaz a vytváří tak další vrstvu nad *LocationManager*. Toto API je k dispozici prostřednictvím komponenty Google Play Services a je doporučováno pro přístup k poloze [3] v systému Android.



Obrázek 1. Schéma LocationManager

Aplikace může pomocí *FusedLocationProviderApi* registrovat žádost o periodické aktualizace polohy ve dvou režimech – aktivním nebo pasivním. Při pasivním režimu dostává aplikace aktualizace polohy při požadavcích od ostatních aplikací v systému. Při aktivním jsou aplikaci doručovány aktualizace v explicitně daném intervalu. Ani v jednom z těchto režimů nemusí být aplikace spuštěna. Pasivní režim nemá vliv na spotřebu energie, kdežto aktivní má podle nastavení dalších proměnných větší či menší vliv na spotřebu energie. Pokud programátor nastavil u aktivního režimu, že žádá o vysokou přesnost polohy (API se pokusí zapnout GPS modul), mají pak další proměnné minimální vliv a je tedy pak toto API velmi energeticky náročné.

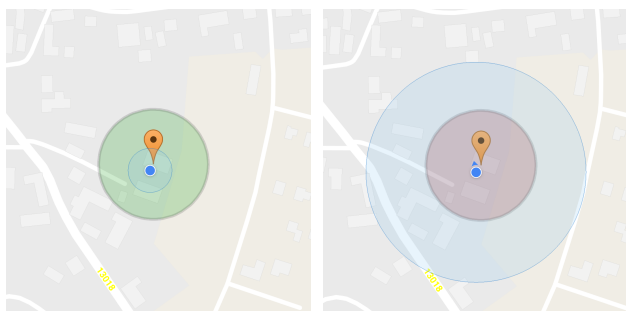
Prostřednictvím Google Play Services je k dispozici také *GeofencingApi* [4]. Díky tomuto API lze zaregistrovat v systému Android tzv. geofence, což jsou kruhové oblasti. Samotný systém pak tyto oblasti hlídá a pokud zařízení do takové oblasti vstoupí, setrvává v ní nebo ji opustí, může dát o této události vědět žádající aplikaci, která ani nemusí být spuštěna. *GeofencingApi* je pro tyto účely optimalizované a využívá aktuálně nejpresnější polohu. Nezapíná GPS modul a má nízkou spotřebu energie. Toto API je navíc z velké části pasivní, takže se snaží využívat žádosti o polohu od jiných aplikací. Pokud jakákoliv aplikace (například widget s počasím) zažádá o polohu, zkontroluje se s touto polohou také stav jednotlivých geofence. Je doporučeno, aby jednotlivé geofence měly minimální poloměr 100 metrů [4].



Obrázek 2. GeofencingApi – geofence a detekce jednotlivých stavů.

3. Experimentování s GeofencingApi a zvýšení jeho spolehlivosti

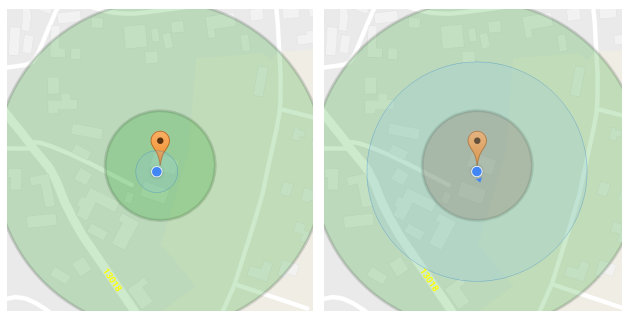
Může se zdát, že GeofencingApi je ideální pro tento typ aplikace – bylo pro takovéto případy užití vytvořeno a optimalizováno. Doporučovaná velikost geofence (100 metrů) je ale pro účely této aplikace příliš velká. Při testování se ukázalo, že opravdu systém detekuje stavy geofence s poloměrem menším než 100 metrů špatně. Aplikace velmi často hlásila, že zařízení opustilo místo a po několika minutách zase vstoupilo na místo i když zařízení bylo stále na stejném místě. Většinu času není na zařízení GPS modul zapnut, proto se poloha určuje pouze pomocí WiFi sítě, která má přesnost nejčastěji okolo 50 metrů. Tato hodnota není konstantní a většinou kolísá mezi hodnotami 50 metrů až 100 metrů. To může způsobit, že geofence s poloměrem 50 metrů systém Android detekuje jako opuštěné, viz obrázek 3.



Obrázek 3. Občasná ztráta přesnosti polohy a následné opuštění místa s poloměrem 50 metrů

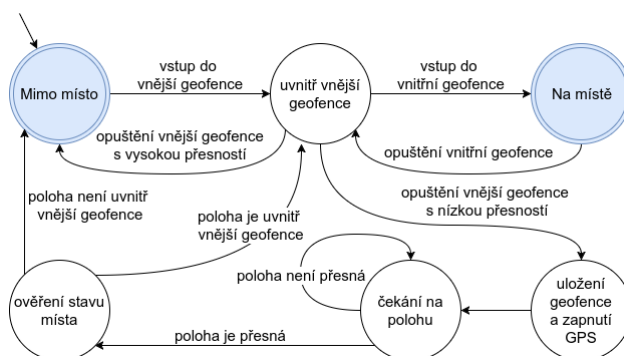
Odstranění tohoto problému bylo vyřešeno tzv. dvoufázovým ověřením. Každé místo má svou původní geofence (minimálně 50 metrů velkou) a druhou vnější geofence s poloměrem o 75 metrů větší (číslo získáno experimentálně). Aby aplikace prohlásila, že zařízení opustilo místo, musí zařízení opustit obě geofence a aby aplikace prohlásila, že zařízení vstoupilo na místo, musí zařízení vstoupit do obou geofence. Pokud zařízení ztratí na přesnosti polohy, opustí se s velkou pravděpodobností pouze vnitřní geofence a zařízení se jeví jako stále na místě. Situace je znázorněna na obrázku 4.

Tento přístup značně zvýšil spolehlivost. Na druhou stranu zvýšil vzdálenost od místa, které zařízení musí urazit, aby aplikace detekovala místo jako opuštěné. Bohužel i tak v menší míře občas aplikace detekovala chybně místo jako opuštěné. Testováním bylo zjištěno, že tento problém způsobuje ztráta signálu WiFi. Stačí nepatrná ztráta signálu nebo přechod na jiný AP (přístupový bod) a zařízení se na okamžik přepne na mobilní síť a získá polohu z tohoto zdroje. Mobilní síť mají přesnost polohy i několik kilometrů



Obrázek 4. Dvoufázové ověření u místa s poloměrem 50 metrů

a proto tady dvoufázové ověření nepomůže. Při detekování události nabízí GeofencingApi možnost získat informaci, jaká byla přesnost polohy při vyvolání události. Pokud tedy při opuštění místa byla přesnost polohy nízká (menší než 75 metrů), uloží aplikace danou geofence a zaregistruje aktualizace polohy pomocí FusedLocationProviderApi v aktivním režimu. To způsobí vyšší spotřebu, ale při získání přesnější polohy může aplikace rozhodnout jestli daná událost byla chybná či ne. Algoritmus je naznačen konečným automatem na obrázku 5.



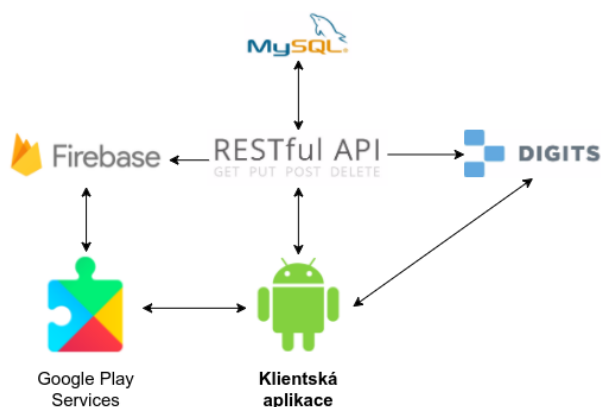
Obrázek 5. Algoritmus automatické detekce přítomnosti na místě

4. Implementace klientské a serverové aplikace

Při programování Android aplikace byla využita architektura Model-View-Presenter. Pro autorizaci uživatele byla využita služba Twitter Digits¹, která dokáže ověřit telefonní číslo zasláním ověřovací SMS. Klientská aplikace je připojena k serverům služby Firebase² prostřednictvím Google Play Services. To umožňuje přijímání *push notifikací*, odesílání chybových hlášení, odesílání analytických informací a generování pozvánky do aplikace. K připojení a využívání služeb Firebase není potřeba, aby aplikace byla spuštěna, protože samotnou komunikaci zajišťují Google Play Services

¹<https://get.digits.com/>

²<https://firebase.google.com/>



Obrázek 6. Architektura komunikace klientské a serverové části

jediným otevřeným kanálem pro všechny aplikace na zařízení využívající služeb Firebase. Potřebuje-li tedy aplikace poslat něco do Firebase, předá to systémové komponentě Google Play Services. Pokud naopak potřebuje Firebase doručit data klientské aplikaci, pošle tato data otevřeným kanálem komponentě Google Play Services, která následně aplikaci probudí a předá jí přijatá data. Klientská aplikace komunikuje se serverovou částí pomocí REST API. Na Androidu je rozhraní definováno knihovnou Retrofit³, která umožňuje jednoduché asynchronní volání. Veškerá komunikace se serverem probíhá pomocí zabezpečeného protokolu HTTPS.

Na serverové části je definováno REST API pomocí PHP frameworku Slim⁴. Samotná data jsou pak uložena v databázi MySQL. Server také zprostředkovává zasílání zpráv pomocí Firebase Cloud Messaging. Každému zařízení přiřadí Firebase jedinečný identifikátor, který musí server uložit do databáze a použít při posílání zpráv jednotlivým zařízením přes Firebase Cloud Messaging. Celá architektura komunikace mezi jednotlivými součástmi je znázorněna na obrázku 6.

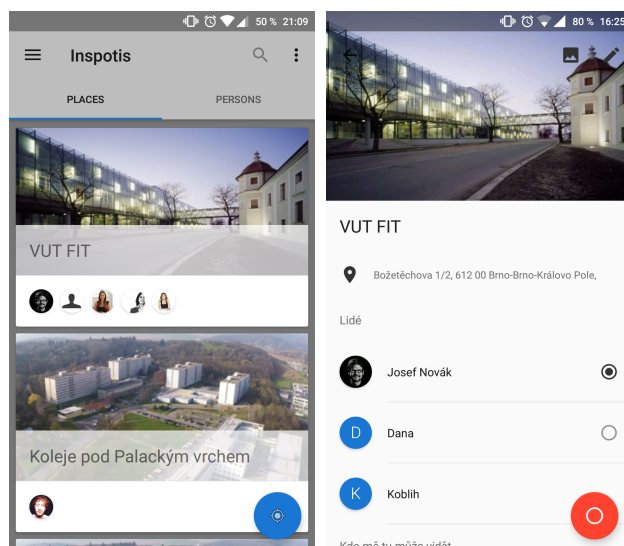
5. Uživatelské prostředí klientské aplikace

Součástí tohoto projektu bylo i navržení a vytvoření uživatelského prostředí tak, aby bylo co nejvíce intuitivní a zároveň dobře plnilo svou funkci. Aplikace je navržena tak, aby tyto případy užití byly pro uživatele nejdostupnější:

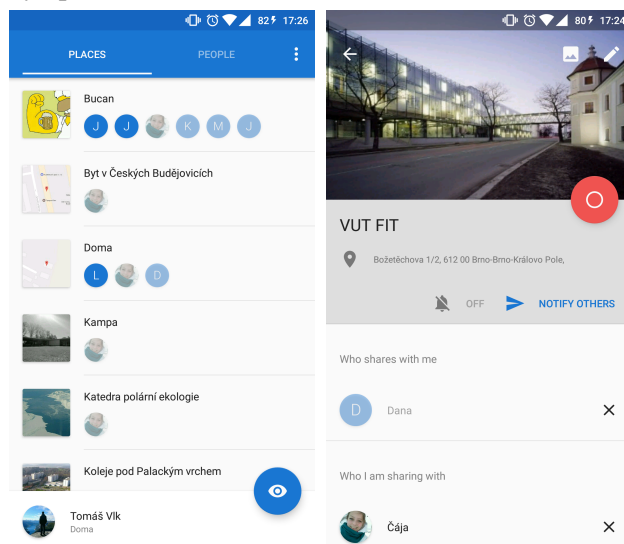
- zjistit, kdo je na daném místě
- zjistit, je-li daná osoba na nějakém místě
- dát vědět určitým osobám svoji přítomnost na daném místě

³<https://square.github.io/retrofit/>

⁴<https://www.slimframework.com/>



Obrázek 7. Jedna z variant uživatelského rozhraní testovaná na uživateli a následně iterativně vylepšovaná.



Obrázek 8. Aktuální podoba uživatelského rozhraní.

Vytvořil jsem několik prototypů, které byly následně testovány na reálných uživateli a sledoval jsem, jak uživatel rozhraní chápe a které úkony mu dělají problémy [5]. Příklad takových prototypů je na obrázku 7. Následně jsem iterativně zlepšoval uživatelské prostředí a opakovaně testoval na uživateli. Příklad toho, jak aplikace aktuálně vypadá, lze vidět na obrázku 8.

6. Závěr

Tento článek popisuje aplikaci pro systém Android, která nabízí uživatelům sdílet jejich přítomnost na určitých místech a přitom držet velkou kontrolu nad svým soukromím.

Aplikace dokáže spolehlivě detekovat zařízení na místech s poloměrem 50 metrů a větším. Odhadovaná spotřeba energie aplikace za 1 den uváděná systémem Android 6.0.1 se pohybuje okolo 20 mAh. Tato hod-

nota byla naměřena na více mobilních zařízeních. GeofencingApi systému Android s mírnými úpravami funguje spolehlivě a není energeticky náročné. Společně se službou Firebase lze vytvořit systém sdílení polohy mezi uživateli bez potřeby aktivní spuštěné součásti aplikace. Vývojáři mohou využít metodu detekce přítomnosti na místě použitou v této aplikaci.

Aktuálně je aplikace zveřejněna a k dispozici zdarma na Google Play Store pod názvem Inspotis. Pokud bude u uživatelů zájem o tuto aplikaci, bude rozšířena na platformu iOS.

Poděkování

Chtěl bych poděkovat prof. Ing. Adamu Heroutovi, Ph.D a všem, kteří mi pomáhali při testování aplikace.

Literatura

- [1] U. Bareth. Simulating power consumption of location tracking algorithms to improve energy-efficiency of smartphones. In *2012 IEEE 36th Annual Computer Software and Applications Conference*, pages 613–622, July 2012.
- [2] Location strategies. *Android Developers*. cit. 29.3.2017.
- [3] Receiving location updates. *Android Developers*. cit. 29.3.2017.
- [4] Creating and monitoring geofences. *Android Developers*. cit. 29.3.2017.
- [5] Steve Krug. *Don't make me think revisited: A common sense approach to web and mobile usability*. Berkeley. CA: New Riders Press, 2014.