

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

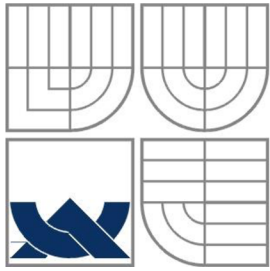
APLIKÁCIA PRE SLEDOVANIE POLOHY ZARIADENIA
PRE OS WINDOWS PHONE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

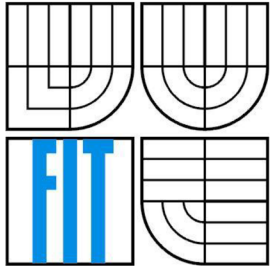
AUTOR PRÁCE
AUTHOR

JURAJ PUPÁK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

APLIKACE PRO SLEDOVÁNÍ POLOHY ZAŘÍZENÍ PRO OS WINDOWS PHONE

APPLICATION FOR TRACKING DEVICES FOR WINDOWS PHONE OS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JURAJ PUPÁK

VEDOUČÍ PRÁCE
SUPERVISOR

Ing. Pavel Očenášek, Ph.D

BRNO 2013

Abstrakt

Aplikace implementována v rámci této práce se zabývá získáváním pozice o zařízení pomocí systému GPS. Aplikace je implementována pro platformu Windows Phone 7.5. Aplikace umožňuje vytvářet profily, které se pak vyhodnocují se získanými pozicemi. Vyhodnocování pozice probíhá na základě dvou bodů pomocí formule Haversine. Teoretická část práce se zabývá technologiemi a systémem GPS, praktická část se zabývá stránkou návrhu a implementace.

Abstract

This application deals with recording of position about phone devices with help of system GPS. The application is implemented for platform Windows Phone 7.5. The Application allows to create profiles, which are evaluated with recorded positions. The evaluation is made through 2 points with Haversine formula. The theoretical part of thesis deals with technologies and GPS system, the practical part deals with design and implementation.

Klíčová slova

GPS, Windows Phone 7.5, Silverlight, C#, Pozice, Profil, Bing Maps, Google Maps Api, Haversine formula

Keywords

GPS, Windows Phone 7.5, Silverlight, C#, Position, Profile, Bing Maps, Google Maps Api, Haversine formula

Citace

Juraj Pupák: Aplikácia pre sledovanie polohy zariadenia pre OS Windows Phone, bakalárska práca, Brno, FIT VUT v Brně, 2013

Aplikácia pre sledovanie polohy zariadenia pre OS Windows Phone

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Juraj Pupák
15.5.2013

Poděkování

Chcel by som poďakovať svojmu vedúcemu Ing. Pavlovi Očenáškovi, Ph.D za rady, ktoré mi poskytol. Takisto by som chcel poďakovať Ing. Rudolfovi Kajanovi, ktorý mi zapožičal mobilný telefón na testovacie účely.

© Juraj Pupák, 2013

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

| | |
|--|----|
| 1 Úvod | 6 |
| 2 Lokalizácia..... | 7 |
| 2.1 Systém GPS | 7 |
| 2.2 Výpočet vzdialenosti medzi dvoma bodmi..... | 8 |
| 3 Technológie | 9 |
| 3.1 Windows Phone 7.5 | 9 |
| 3.2 Silverlight | 11 |
| 3.3 WCF..... | 11 |
| 3.4 .Net Framework | 12 |
| 3.5 Windows Phone Developer Tools | 12 |
| 4 Ochrana osobných údajov..... | 13 |
| 4.1 Ochrana osobných údajov detí..... | 13 |
| 4.2 Ochrana osobných údajov zamestnanca..... | 14 |
| 5 Analýza existujúcich riešení | 15 |
| 5.2 Mobile Spy..... | 15 |
| 5.2 Mobistealth Mobile Phone Spy..... | 16 |
| 6 Návrh | 17 |
| 6.1 Funkcionalita | 17 |
| 6.2 Diagramy prípadu použitia | 18 |
| 6.3 Architektúra klient-server | 19 |
| 7 Implementácia..... | 26 |
| 7.1 Databáza | 26 |
| 7.2 WCF Servis..... | 28 |
| 7.3 GpsBrowser | 29 |
| 7.4 GpsMonitor..... | 31 |
| 8 Testovanie..... | 38 |
| 9 Záver..... | 39 |
| Literatúra | 40 |
| Zoznam príloh..... | 41 |
| Príloha A. Obsah CD | 42 |

Kapitola 1

Úvod

V dnešnej dobe sa stali mobilné telefóny neoddeliteľnou súčasťou nášho každodenného života. Od počiatku, kedy sa mobilné telefóny využívali len na telefonovanie a písanie textových správ, sme sa dostali do doby, kedy nám mobilný telefón slúži nielen na telefonovanie a písanie, ale aj na prezeranie internetu, poriadkovanie obrazových a zvukových záznamov, prezeranie videí a dokonca poskytujú aj modul GPS. Vďaka tomuto môže byť mobilný telefón využitý ako navigácia do automobilov alebo ako navigácia pri turistike.

Cieľom tejto bakalárskej práce je vytvoriť aplikáciu, ktorá bude získavať pomocou GPS modulu pozície o aktuálnej polohe mobilného telefónu. Aplikácia bude taktiež vyhodnocovať jednotlivé pozície s profilmi a informovať užívateľa v prípade ich porušenia. Aplikácia bude navrhnutá tak, aby bola prijateľná pre každú vekovú skupinu s využitím technológií, ktoré Windows Phone 7.5 ponúka.

Bakalárska práca je členená do viacerých kapitol. Princíp a história systému GPS sú obsiahnuté v kapitole 2. Kapitola 3 popisuje jednotlivé technológie od Windows Phone, cez Silverlight až po .Net Framework. Práca sa zaoberá aj problematikou osobných údajov v kapitole 4. Kapitola 5 popisuje existujúce riešenia. V kapitole 6 je popísaný priebeh návrhu aplikácie. Implementačná časť sa nachádza v kapitole 7. Predposlednou kapitolou je kapitola 8, ktorá sa zaoberá testovaním aplikácie v reálnom prostredí a poslednou kapitolou je kapitola 9, kde je záver a zhrnutie celej práce.

Kapitola 2

Lokalizácia

Najpodstatnejšou časťou celej bakalárskej práce bolo lokalizovanie zariadenia. K tomuto účelu poslúžil systém GPS. Samotné zistenie polohy ale nebolo postačujúce, nakoľko bolo potrebné získanie vzdialenosti medzi dvomi bodmi, čo slúžilo na vyhodnotenie porušenia alebo neporušenia časového profilu. V tejto kapitole vás oboznámim so systémom GPS, jeho históriou a určením polohy medzi dvoma bodmi.

2.1 Systém GPS

Globálny systém určenia polohy je satelitný navigačný systém na určenie polohy zariadenia nachádzajúceho sa na Zemi. História tohoto systému sa začína od začiatku 70. – tich rokov minulého storočia. Hlavnou myšlienkou bolo vybudovať družicový, pasívny systém, pomocou ktorého by bolo možné určiť presnú polohu telesa spolu s presným časom v trojrozmernom priestore. Projekt bol oficiálne pomenovaný NAVSTAR – GPS. Prevádzkovateľom tohto systému, Ministerstva obrany USA bolo, aby vojenské jednotky mohli presne určiť polohu a rýchlosť. Z toho vyplýva, že projekt bol hlavne určený na vojenské účely. Od 90. rokov bol tento systém sprístupnený aj pre civilné účely, ale s obmedzením presnosti polohy na 100m z dôvodu bezpečnosti. Po roku 2000 bol systém plnohodnotne sprístupnený, čo znamenalo veľké výhody.[1]

Družicové polohové systémy sa delia na 3 základné segmenty[1]:

Kozmický segment

- Je tvorený sústavou družíc obiehajúcich okolo Zeme. GPS má 24 družíc, z toho 21 družíc je pracovných a 3 sú rezervné. Družice sú umiestnené vo vzdialenosti 20183km na povrchom Zeme a sú umiestnené na šiestich obežných dráhach.

Riadiaci segment

- Tvoria pozemné stanice. Hlavnou riadiacou stanicou je stanica na leteckej základni v Colorado Springs v USA. Ostatné monitorovacie stanice sú

rozmiestnené rovnomerne po obvodě Zeme okolo rovníka. Riadiaci systém vykonáva správu a údržbu družíc.

Užívateľský segment

- Skladá sa z GPS prijímačov jednotlivých užívateľov a iných technických zariadení, ktoré umožňujú využívanie polohového systému. Nakoľko prijímače nekomunikujú s družicami, je systém GPS schopný obslúžiť neobmedzený počet užívateľov.

Výhodou systému GPS je to, že je úplne bezplatný. Nevýhodou je potreba priamej viditeľnosti na nebo. K dispozícii je treba mať minimálne 4 družice. Z toho dôvodu nejde merať vzdialenosť pod vodou, v tuneloch a bytových priestoroch.[1]

2.2 Výpočet vzdialenosti medzi dvoma bodmi

Nakoľko bolo potrebné zaznamenávať vzdialenosť zariadenia od vytýčeného bodu v danom profile, bolo potrebné nájsť vhodnú metódu na určenie vzdialenosti medzi dvoma bodmi. Pre tieto účely postačuje metóda Haversine.[2]

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2\left(\frac{\Delta\lambda}{2}\right)$$
$$c = 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$
$$d = R * c$$

φ – zemepisná šírka

λ – zemepisná dĺžka

R – polomer Zeme(6371km)

Kapitola 3

Technológie

3.1 Windows Phone 7.5

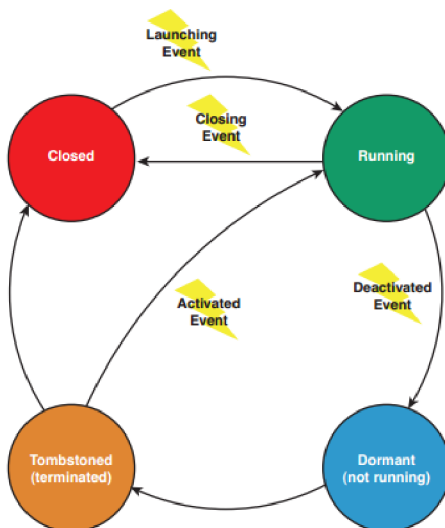
OS Windows Phone je priamym nástupcom OS Windows Mobile. Systém sa začal vyvíjať už v roku 2004 a jeho vtedajšie označenie bolo „Photon“. Žiaľ, projektu sa veľmi nedarilo a jeho vývoj bol ukončený. O pár rokov neskôr sa firma Microsoft pokúsila o jeho znovuzrodenie. Na vývoji sa začal podieľať reorganizovaný vývojový tím a vývoj začal na plné obrátky. V roku 2010 bol OS Windows Phone uvedený na trh a predstavený na mobilných zariadeniach od spoločností ako: Dell, HTC, Samsung, LG. Na trh bola uvedená verzia Windows Phone 7. Po vydaní aktualizácii prišiel na trh Windows Phone 7.5 pod názvom Mango, ktorý doplnil niektoré chýbajúce veci a opravil tie staré. Užívateľské rozhranie Windows Phone je založené na systéme Metro¹. Úvodnú obrazovku tvoria tzv. „dlaždice“. Dlaždice sú odkazmi na aplikácie, funkcie, individuálne položky. Sú dynamicky meniace sa v reálnom čase – napr. dlaždica pre emailové konto zobrazuje stav neprečítaných emailov.[3] Ďalšou novou vlastnosťou užívateľského rozhrania je systém panoramatických obrazoviek. Vytvára dynamické aplikácie umožňujúce väčšie množstvo operácií nad jednou obrazovkou.[4]



Obrázok 1. Panoramatická obrazovka

¹ Metro – jazyk určený na tvorbu dizajnu od spoločnosti Microsoft

Windows Phone prichádza s podporou multitaskingu². Multitasking je riešený pomocou procesov bežiacich na pozadí. Medzi procesy bežiace na pozadí patria: naplánované úlohy, notifikácie ako napr. pripomienkovač, budík.



Obrázok 2. Životný cyklus aplikácie[5]

OS Windows Phone poskytuje užívateľom dve výkonné technológie na tvorbu aplikácií:

- Silverlight
- XNA³

Obe technológie sú v podstate rozdielne – každá bola vytvorená za iným účelom. Silverlight je preferovaná technológia na tvorbu biznis aplikácií. XNA ponúka vysoký grafický výkon 2D a 3D s podporou shaders, detekcie kolízií, pozičným zvukom. V súčasnej dobe sú jazyky C# a VB.Net podporované ako pre Silverlight tak aj pre XNA.[5]

| Funkcie | XNA | Silverlight |
|----------------|------------|--------------------|
| Web | Nie | Áno |
| Desktop | Áno | Áno |
| Zune | Áno | Nie |
| Xbox | Áno | Nie |
| 2D | Áno | Áno |
| 3D | Áno | Nie |
| Xbox LIVE | Áno | Nie |
| Rich Controls | Nie | Áno |

² Multitasking – vykonávanie viacerých úloh súčasne

³ XNA – technológia na tvorbu video hier

| | | |
|---|-----|-----|
| Web browser control | Nie | Áno |
| Vector graphic | Nie | Áno |
| Layout support | Nie | Áno |
| Free design-time environment | Nie | Áno |
| Advanced tools support(physics, shader) | Áno | Nie |
| Data binding and styles | Nie | Áno |
| GPU acceleration | Áno | Áno |
| Full screen video | Áno | Áno |
| Composite video | Nie | Áno |
| Hardware keyboard support | Áno | Áno |
| Software keyboard support | Nie | Áno |

Tabuľka 1. Porovnanie technológií[5]

3.2 Silverlight

Technológia Silverlight bola primárne vytvorená pre tvorbu webových aplikácií. Je založená na technológii Windows Presentation Foundation(WPF) a po prvý krát bola uvoľnená v roku 2007. Silverlight v sebe zahŕňa multimédiá, grafiku(vektorovú) ai. Užívateľské prostredie je definované v jazyku XAML⁴, ktorý je založený na formáte XML. O technológii Silverlight možno hovoriť ako o konkurentovi známeho a dlho používaného Adobe Flash. Veľkou výhodou Silverlightu je to, že je založený na zredukovanej verzii spoločného runtime jazykov .Net, čo umožňuje vytvárať veci klienta v čistom C#. Silverlight je tzv. odľahčený zásuvný modul pre webové prehliadače.[6]

3.3 WCF

WCF je jednou z ďalších technológií z dielne Microsoft, ktorá je využitá v tejto aplikácii. Technológia WCF je framework, ktorý slúži na tvorbu servisne orientovaných aplikácií. Umožňuje posielat' dáta ako asynchrónne správy z jedného koncového bodu na druhý. Koncový bod môže byť nejaký servis, ktorý beží na IIS⁵ alebo to môže byť servis v rámci aplikácie.[7] Pred WCF technológiou bolo možné vytvárať aplikácie komunikujúce po sieti pomocou nasledujúcich technológií:

1. ASP.NET Web Services
2. Web Services Enhancements
3. .NET Remoting

⁴ XAML - <http://msdn.microsoft.com/en-us/library/cc295302.aspx>

⁵ IIS – webový server

4. Enterprise Services
5. Microsoft Message Queuing

Výber danej technológie spočíval v tom, aké boli kladené požiadavky na danú aplikáciu. Nevýhodou boli horšia údržba a rozvoj takejto aplikácie. Práve toto bolo podnetom pre vytvorenie WCF, ktorý všetky hore uvedené technológie zapúzdruje a tým umožňuje vytvárať aplikácie jednoduchšie na údržbu a ich ďalší rozvoj.[8]

3.4 .Net Framework

.Net Framework je technológia, ktorá bola vyvinutá pre lepšie vytváranie programov hlavne pre Windows. Prvá verzia bola vydaná v roku 2002 a bola veľkým prielomom ako v oblasti technologickej, tak aj v oblasti programátorskej. Spolu s .Net Framework bol vydaný aj nový programovací jazyk C#. V dnešnej dobe je dostupná verzia .Net 4.5. Technológia .Net prekladá všetky programy napísané v rôznych programovacích jazykoch do medzikódu CIL (Common Intermediate Language). Takýto kód je potom uložený do assembly súborov. Po spustení týchto súborov je na ne aplikovaný CLR (Common Language Runtime), ktorý dané súbory preloží do natívneho kódu procesoru.[9]

3.5 Windows Phone Developer Tools

S príchodom operačného systému Windows Phone prišla aj sada nástrojov na tvorbu aplikácií, ktorá mala uľahčiť prácu vývojárom. Táto sada obsahuje veľké množstvo funkcií, ktoré možno využiť. Windows Phone Developer Tools je voľne dostupná⁶. Momentálne je dostupná verzia pre Windows Phone 8, ale tá vyžaduje mať nainštalovaný operačný systém Windows 8. Súčasťou týchto nástrojov je aj emulátor Windows Phone Emulator, ktorý kvázi nahrádza reálne zariadenie. Neponúka ale kompletnú funkcionálnosť ako reálne zariadenie, ale na účely vývoja aplikácií je postačujúce. Emulátor nemá priamo v sebe zabudovaný GPS modul, ale je možnosť využitia Additional Tools, ktoré sú súčasťou Windows Phone Developer Tools a tým dokážu nahradiť chýbajúci GPS modul.

⁶ <http://www.microsoft.com/en-us/download/details.aspx?id=13890>

Kapitola 4

Ochrana osobných údajov

Sledovanie a monitoring činností človeka je veľmi citlivou a preberanou otázkou. Každý človek má právo na súkromie a akýkoľvek zásah do súkromia môže byť v rozpore so zákonom a zneužitie osobných údajov môže byť trestné⁷.

Keďže je aplikácia vytváraná za účelom sledovania polohy detí a zamestnancov, bolo potrebné zistiť, aké obmedzenia platia v oboch prípadoch. V oboch prípadoch sa vychádza zo zákona ČR.

4.1 Ochrana osobných údajov detí

K základným princípom patrí:

1. povinnosť rešpektovať prednostný záujem dieťaťa, čo sa premieta do primeranosti spracovania údajov vo vzťahu k účelu,
2. povinnosť chrániť blaho dieťaťa a starať sa oň,
3. oprávnenie nepľnoletého sa rozširuje podľa vývojového stupňa, s čím súvisí to, že spracovanie osobných údajov musí byť v korelácii s vývojom dieťaťa,
4. oprávnenie dieťaťa byť konzultovaný (vypočuť si jeho názor a vziať ho do úvahy), prípadne sa rozhodovať sám pod dozorom zákonného zástupcu.

Lokalizáciou možno určiť, kde sa subjekt údajov približne nachádza bezohľadu nato, či informácia o mieste je poskytnutá prostým zobrazením na mape alebo je vyjadrená inak. Aktuálny alebo inak časovo určený výskyt sledovaného subjektu údajov na určitom mieste môže byť vyjadrený graficky alebo inak a spojený s ďalšími informáciami. Lokalizácia môže byť v rozpore so slobodou anonymného pohybu.

Problémom môže byť, pokiaľ rodičia či iní zákonní zástupcovia používajú k lokalizácii detí jeho mobilný telefón, bez súhlasu detí. Takýto postup je značným zásahom do súkromia a je nezákonný. Úrad pre ochranu osobných údajov nemôže toto konanie dozorovať, prípadne

⁷ http://www.oou.cz/index.php?file=personal_data_protection_zakon_101

sankcionovať, lebo z pohľadu zákona o ochrane osobných údajov sa jedná o spracovanie osobných údajov pre vlastnú potrebu.[10]

4.2 Ochrana osobných údajov zamestnanca

Zamestnávateľ ako správca osobných údajov môže podľa § 5 odst. 2 zákona č. 101/2000 Sb., o ochrane osobných údajov a o zmene niektorých zákonov, spracovávať osobné údaje len so súhlasom subjektu údajov. Zamestnanci nesmú bez súhlasu zamestnávateľa užívať pre svoju vlastnú potrebu výrobné a pracovné prostriedky zamestnávateľa vrátane výpočtovej techniky a jeho telekomunikačných zariadení. Zamestnávateľ nesmie bez závažného dôvodu spočívajúceho vo zvláštnej činnosti zamestnanca narušovať súkromie zamestnanca na pracovisku a v spoločných priestoroch zamestnávateľa tým, že zamestnanca podrobuje otvorenému alebo skrytému sledovaniu, odposluchu a záznamu jeho telefonických hovorov. Ak je u zamestnávateľa daný závažný dôvod spočívajúci vo zvláštnej činnosti zamestnanca, ktorý odvodňuje zavedenie kontrolných mechanizmov, je zamestnávateľ povinný priamo informovať zamestnanca o rozsahu kontroly a spôsoboch ich vykonávania.[11]

Kapitola 5

Analýza existujúcich riešení

Zo zadania vyplývajú určité požiadavky, ktoré by mala aplikácia splňovať, a ktoré bolo treba zanalyzovať. Najlepším riešením je analýza už existujúcich riešení podobného typu, zistiť výhody a nevýhody týchto riešení. Táto kapitola sa zaoberá analýzou existujúcich riešení, funkčnými. Na analýzu poslúžili dve aplikácie, z ktorých sa potom vychádza pri návrhu aplikácie. Obe aplikácie sú spolpatnené, ale je možnosť vyskúšať skúšobnú verziu.

5.2 Mobile Spy

Mobile Spy Smartphone Monitoring Software⁸ je od spoločnosti Retina-X Studios. Aplikácia je určená pre platformy: iPhone, Android, BlackBerry a Symbian. Platforma Windows Phone nie je podporovaná. K svojej činnosti využíva pripojenie k internetu, prostredníctvom ktorého odosiela zaznamenané aktivity na server, resp. na Mobile Spy účet, ktorý je k dispozícii po zakúpení produktu. Na servery je si možné zobrazit' štatistiky jednotlivých zaznamenaných údajov.

Medzi jeho funkcie patrí:

- monitorovanie sms správ,
- chat monitoring: facebook, twitter,
- zaznamenávanie prichádzajúcich a odchádzajúcich hovorov,
- zobrazenie fotiek, videí na telefóne,
- zaznamenávanie GPS lokácie,
- možnosť Live Screen.

⁸ <http://www.mobile-spy.com/>

5.2 Mobistealth Mobile Phone Spy

Aplikácia Mobistealth Mobile Phone Spy⁹ poskytuje tzv. All-In-One riešenie, tj. umožňuje monitorovať ako aj deti, tak aj správanie zamestnancov ako nakladajú so služobným telefónom.

Funkcie:

- zaznamenať reálnu a minulú polohu,
- nahrávanie hovorov,
- monitorovanie okolia naživo, odpočúvanie okolia,
- história webu,
- zmeny týkajúce sa SIM.

Aplikácia beží na pozadí a užívateľ o nej nevie, pretože aplikácia je pred ním schovaná. Opäť ako aj v predošlej aplikácii, aj tu je k dispozícii účet na servery, kde sa zhromažďujú zaznamenané dáta, ktoré sú rozčlenené do jednotlivých kategórií(SMS, Volania, Kontakty, atď.).

Ako možno vidieť, obe spomínané aplikácie pracujú systémom klient – server, kde server slúži ako úložisko pre zaznamenané dáta, ktoré je možno prezerat' z internetového prehliadača.

⁹ <http://www.mobistealth.com/mobile-phone-spy-software>

Kapitola 6

Návrh

Po preštudovaní a vyskúšaní existujúcich riešení prichádza na rad návrh aplikácie. Hlavným cieľom bakalárskej práce bolo navrhnúť a implementovať aplikáciu, ktorá bude zaznamenávať polohu zariadenia a vyhodnocovať polohu podľa vopred nastavených časových profilov. Ďalšou požadovanou funkcionalitou bolo, aby aplikácia zaznamenávala činnosť užívateľa na mobilnom telefóne, tj.: zaznamenávanie prichádzajúcich/odchádzajúcich hovorov, prijaté/odoslané správy, celkovo záznam o činnosti vykonávanej na mobilnom telefóne. Podľa odborných článkov, rôznych odborných fór došlo k zisteniu, že nemožno implementovať túto požadovanú funkcionalitu z toho dôvodu, že Microsoft neposkytuje vývojárom verejné API¹⁰, ktoré je potrebné na implementáciu takejto funkcionality. Microsoft sa rozhodol urobiť tento krok z dôvodu ochrany osobných údajov a bezpečnosti¹¹. Celý návrh aplikácie je výlučne zameraný len na lokalizovanie polohy zariadenia a jej vyhodnocovanie s vopred stanovenými časovými profilmi – GpsMonitor. Návrh zahrňuje aj druhú mobilnú aplikáciu, ktorá slúži na prezeranie zaznamenaných polôh - GpsBrowser.

6.1 Funkcionalita

Medzi základné funkcie patria:

- aplikácia je funkčná na mobilnom telefóne s dotykovou obrazovkou a operačným systémom Windows Phone 7.5 Mango,
- aplikácia umožňuje vytvárať časové okná(profily),
- je zabezpečené sledovanie polohy pomocou GPS modulu,
- notifikácia užívateľa o porušení časového profilu prostredníctvom emailu,
- ukladanie získaných údajov na server s možnosťou ich prehliadania,
- možnosť ukladať údaje v podobe offline súboru,
- zobrazenie polohy na mape,
- odosielanie požiadaviek na získanie polohy konkrétneho zariadenia.

¹⁰ API – rozhranie pre programovanie aplikácií, s veľkým množstvom funkcií a tried

¹¹ <http://social.msdn.microsoft.com/Forums/is/wpdevelop/thread/4833b9b9-c477-403d-b289-e4e1ff79cc7b>

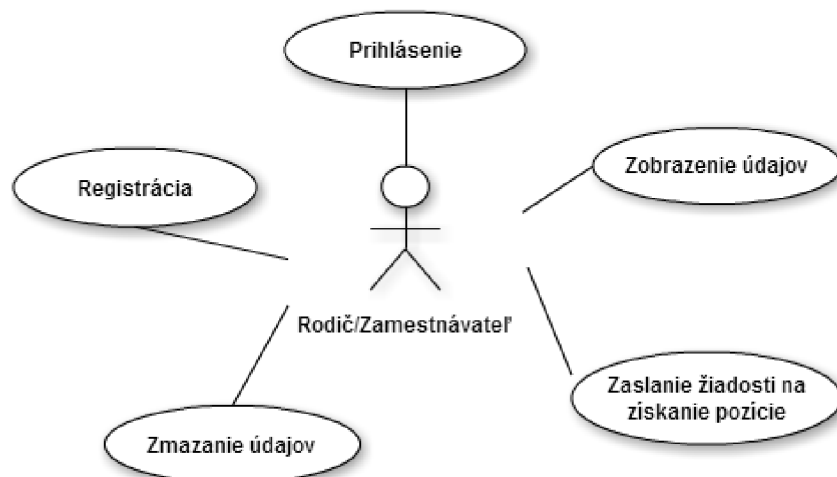
6.2 Diagramy prípadu užitia

Diagram prípadu užitia na obrázku č. 3 znázorňuje, v akých možných prípadoch môže užívateľ aplikáciu GpsMonitor používať.



Obrázok 3. Diagram prípadu užitia GpsMonitor

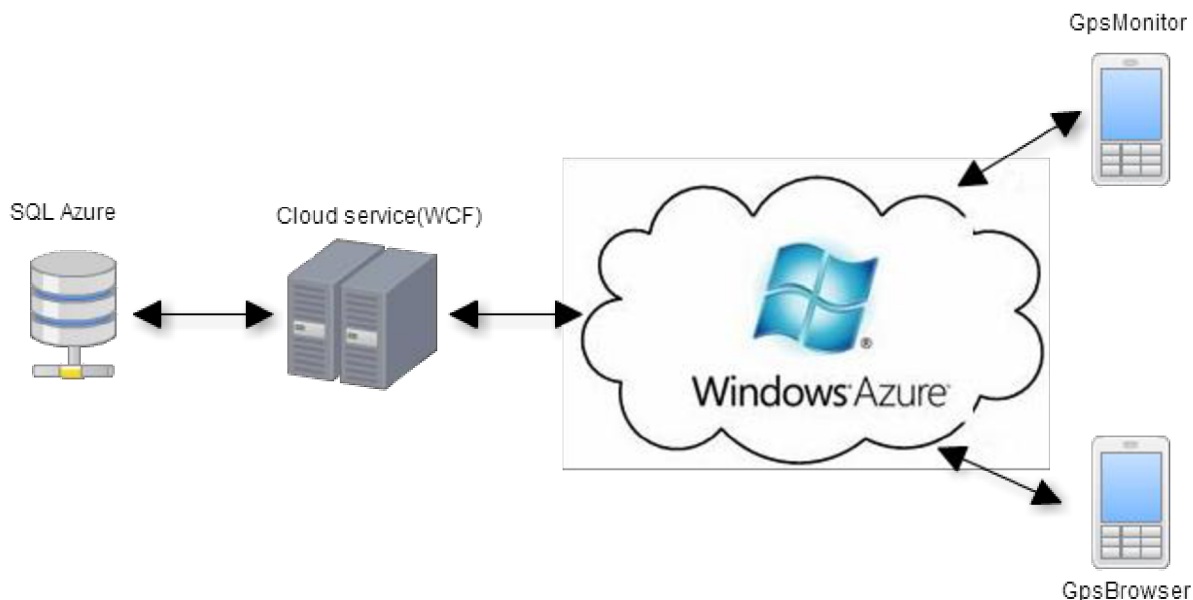
Nasledujúci diagram prípadu užitia znázorňuje v akých možných stavoch možno používať aplikáciu GpsBrowser.



Obrázok 4. Diagram prípadu užitia GpsBrowser

6.3 Architektúra klient-server

V dnešnej dobe je už celkom bežné, že väčšina aplikácií funguje na princípe klient-server. Tak ako aplikácia GpsMonitor, tak aj aplikácia GpsBrowser pracujú na takomto type architektúry. Na obrázku č. 5 je znázornená komunikácia.



Obrázok 5. Schéma komunikácie

6.3.1 GpsBrowser

GpsBrowser slúži na registráciu užívateľa a jeho prihlasovanie. Ponúka možnosť prezerania a mazania zaznamenaných pozícií, ktoré sú umiestnené na servery. Ďalšou funkciou je zaslanie žiadosti na získanie pozície zariadenia. Prihlasovacie údaje užívateľa sú uložené v nastaveniach telefónu – `ApplicationSettings`. Heslo nie je uložené ako bežný text, ale je z neho vytvorený hash¹², aby bolo znemožnené jeho prečítanie v prípade nejakej bezpečnostnej hrozby. Dôvod, pre ktorý sú tieto informácie uložené je ten, že slúžia na vytvorenie šifrovacieho kľúča, pomocou ktorého sú dáta, ktoré sú umiestnené na servery v zašifrovanej podobe, dešifrované. Využitie je pritom symetrické šifrovanie AES. Aplikácia vyžaduje internetové pripojenie, pretože je nutné na komunikáciu so serverom.

6.3.2 GpsMonitor

GpsMonitor je navrhnutý tak, aby mohol pracovať aj v režime bez internetového pripojenia. Návrh vychádza z toho, že nie každý môže mať pripojenie na internet. Aplikáciu bolo nutné ošetriť nejakým heslom, aby nemohol do nej hocikto prístupit' a meniť v nej nastavenia. Heslo sa ukladá do

¹² Hash - <http://sk.wikipedia.org/wiki/Hash>

nastavení aplikácie a je v zašifrovanej podobe, aby ho nebolo možné prečítať, pokiaľ by sa niekto pokúšal heslo zistiť.

GpsMonitor umožňuje nasledovné:

- vytvorenie profilu,
- nastavenie aplikácie,
- prezeranie offline zaznamenaných pozícií.

V nasledujúcich častiach je popísaný návrh vyššie uvedených funkcií.

Vytvorenie profilu

Vytvorenie časového profilu spočíva v nasledovnom:

- Nastavenie mena profilu.
- Nastavenie adresy – počiatočný bod pre konkrétny profil. Adresu je možné vyhľadať alebo označiť konkrétny bod na mape. Toto nastavenie vyžaduje pripojenie na internet.
- Nastavenie vzdialenosti – toto nastavenie slúži k ohraničeniu platnosti profilu od počiatočnej adresy. Minimálna možná hodnota je 150m. Pri prekročení nastavenej vzdialenosti je zaznamenaná pozícia označená ako porušená s príslušným názvom profilu, ktorý bol porušený.
- Nastavenie módu platnosti:
 - Once – užívateľ vyberie dátum a čas od/do.
 - Daily – tento mód určuje každodennú platnosť profilu v určitú hodinu.
 - Weekly – pri výbere tohoto módu má užívateľ možnosť vybrať, v ktoré dni nadobúda profil platnosť. Potrebné je zadať čas od/do.
- Povolenie profilu – určuje, či sa má daný profil brať do úvahy pri vyhodnocovaní

Profily sú ukladané do dátovej štruktúry typu súbor. Formát súboru je v tvare xml¹³. Ako je možné vytvárať profily, tak je možnosť jednotlivé profily aj zmazať. Pri návrhu sa brala do úvahy možnosť ukladať jednotlivé profily na server, pričom by bolo možné vytvárať ich vzdialene. Aplikácia umožňuje pracovať v režime bez pripojenia na internet, tj. zaznamenaná pozícia sa môže vyhodnotiť s niektorým z profilov uložených lokálne, takže vytvárať profily vzdialene by nemalo význam. Ak by boli profily uložené na servery a aplikácia by nemala prístup na internet, nebolo by možné vyhodnocovať jednotlivé pozície s profilmi.

¹³ <http://www.w3.org/XML/>

Nastavenie aplikácie

Nastavenie aplikácie je najdôležitejšou časťou. V nastaveniach sa nachádzajú: emailová adresa, zapnutie/vypnutie monitorovania, vytvorenie/odstránenie odkazu na zariadenie.

Emailová adresa slúži ako kontakt, na ktorú je v prípade porušenia profilu zaslaný email. Zaslanie emailu si vyžaduje internetové pripojenie. Počas návrhu bolo zamýšľané upozorniť užívateľa pri porušení profilu aj iným spôsobom ako len emailom, ktorý potrebuje internet. Jednou z možností bolo zaslanie sms správy bez interakcie užívateľa. Išlo by o variantu, ktorá by bola využitelná takmer za každých okolností. Z bezpečnostných dôvodov to možné ale nie je. Odoslanie správy by bolo možné implementovať, ale len takým spôsobom, že užívateľ by musel potvrdiť odoslanie, čo by v prípade porušenia profilu postrádalo zmysel, aby užívateľ s jeho vedomím odosielať sms o tom, že porušil profil. Ak by bola aplikácia robená pre OS Android, tak takáto možnosť by bola realizovateľná.

Vytvorenie odkazu je dôležitou časťou, pokiaľ sa užívateľ rozhodne, aby zaznamenané pozície boli ukladané na server alebo pokiaľ chce presunúť pozície z telefónu na server. Aby mohol byť vytvorený odkaz na zariadenie, musí mu predchádzať registrácia užívateľa, ktorú je potrebné vykonať v aplikácii GpsBrowser. Okrem prihlasovacích údajov je potrebné zadať jednoznačný názov zariadenia, ktorý ešte nemá užívateľ zaregistrovaný. Po vytvorení odkazu sú použité prihlasovacie údaje uložené v nastaveniach aplikácie, pričom z hesla je vytvorený hash. Potreba ukladať tieto informácie je z toho dôvodu, že slúžia na vytvorenie šifrovacieho kľúča, podľa ktorého sú dáta na servery šifrované. Rovnaký scenár platí aj v aplikácii GpsBrowser, kde tieto informácie slúžia k tomu, aby mohli byť dáta zo serveru dešifrované. Ukladanie šifrovaných údajov na server je z prostého dôvodu a to takého, aby napr. správca databázy nemohol prečítať jednotlivé zaznamenané údaje o osobe a zneužiť ich vo svoj prospech. V prípade zabudnutia mena alebo hesla, sú všetky údaje stratené, pretože neexistuje možnosť ako ich dešifrovať. Ak sa rozhodne užívateľ odkaz na zariadenie odstrániť, tak zariadenie nie je trvalo odstránené z databázy, len mu je nastavený príznak, že nie je naň vytvorený odkaz.

To aby bolo možné vôbec zaznamenávať nejaké pozície, je potrebné zapnúť monitorovanie. Zapnutím monitorovania sa vytvorí nová úloha, ktorá sa vykonáva na pozadí. Táto úloha sa spúšťa raz za 30 minút. Je to východzie nastavenie systému Windows Phone, takže nie je možné ovplyvniť interval spúšťania. Úloha sa vykonáva maximálne 25-30 sekúnd. Opäť je to obmedzenie systému. V rámci tejto úlohy sa získa pozícia. Pozícia nie je reálna, pretože úlohy ktoré pracujú na pozadí nedokážu získať aktuálnu reálnu pozíciu. Namiesto toho sa získa pozícia, ktorá je uložená v pamäti cache. Pozícia je aktualizovaná každých 15 minút. Po získaní pozície sa pozícia vyhodnotí s profilmi a uloží sa buď na zariadenie alebo je odoslaná na server.

Prezeranie pozícií

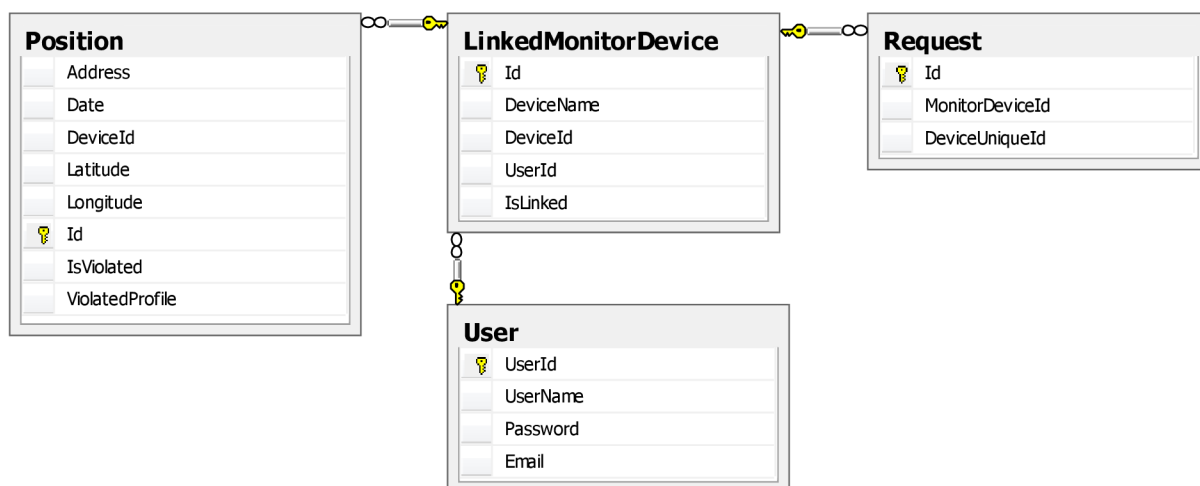
Pokiaľ neexistuje odkaz na zariadenie, tak jednotlivé pozície sú ukladané priamo na zariadenie. Pozície sa ukladajú do dátovej štruktúry súbor, ktorý má formát xml. Zaznamenaná pozícia obsahuje dátum a súradnice. Jednotlivé pozície je možné prezerat' na mape, čo ale vyžaduje pripojenie na internet. Užívateľ môže pozície buďto zmazať alebo ich môže nahrat' na server. V tom prípade musí ale vytvorit' odkaz na zariadenie.

6.3.3 Server

Server je druhou časťou aplikácie. Návrh takéhoto riešenia vychádza z praxe, kde už väčšina súčasných aplikácií pracuje na princípe klient-server. Komunikácia medzi klientom a serverom prebieha prostredníctvom servisu, ktorý je vytvorený pomocou komunikačnej technológie WCF od spoločnosti Microsoft. Ďalšou časťou, ktorá je súčasťou serveru je SQL databáza. Pôvodne mala aplikácia používať len lokálne úložisko na mobilnom telefóne, ale táto možnosť nie je vyhovujúca, preto sa použije databáza umiestnená na servery. V tom prípade bolo potreba poriešiť otázku bezpečnosti aby dáta, najmä zaznamenané pozície, nemohli byť zneužit' v niečí prospech. Preto sú záznamy do databázy ukladané v šifrovanej podobe pomocou symetrického šifrovania AES. Šifrovanie zabezpečuje WCF servis. Servis zabezpečuje komunikáciu s databázou a taktiež sa stará o: registráciu užívateľov, prihlásenie a overenie identity užívateľa, vytvorenie odkazu na zariadenie. Databáza a súčasne aj servis sú umiestnené na systéme Windows Azure¹⁴.

Návrh databázy

Databáza pozostáva zo 4 tabuliek, ktoré je možné vidieť na obrázku č. 6.



Obrázok 6. Diagram SQL databázy

¹⁴ Windows Azure - <http://www.microsoft.com/slovakia/azure/>

Tabuľka User

Tabuľka `User` je tabuľkou, do ktorej sa ukladajú užívatelia, ktorí sa zaregistrovali. Pozostáva zo 4 stĺpcov: `UserId` je primárnym kľúčom tabuľky, `UserName` meno užívateľa, `Password` predstavuje heslo užívateľa v podobe hashu a posledným stĺpcom je `Email`, kde je uložený email užívateľa a má len informatívny charakter.

LinkedMonitorDevice

Nasledujúcou tabuľkou je tabuľka `LinkedMonitorDevice`. Táto tabuľka slúži na ukladanie zariadení, pre ktoré bol vytvorený odkaz. S tabuľkou `User` je vo vzťahu 1:M, tj. užívateľ môže mať prilinkovaných neobmedzený počet zariadení. Tabuľka obsahuje 5 stĺpcov: opäť `Id` ako primárny kľúč, `DeviceName` je meno zariadenia, unikátne číslo, ktoré má každé zariadenie obsahuje stĺpec `DeviceId`, `IsLinked` označuje, či na dané zariadenie existuje odkaz, `UserId` je cudzím kľúčom do tabuľky `User`.

Position

Tabuľka `Position` uchováva jednotlivé pozície prilinkovaných zariadení. Údaje v tejto tabuľke sa nachádzajú v šifrovanej podobe, okrem stĺpcov: `Id`, `Date`, `IsViolated`, `ViolatedProfile`. Stĺpce: `Address`, `Latitude`, `Longitude` sú zašifrované z dôvodu bezpečnosti. Stĺpec `DeviceId` je cudzím kľúčom do tabuľky `LinkedMonitorDevice`.

Request

Poslednou tabuľkou je tabuľka `Request`. Táto tabuľka je vo vzťahu s tabuľkou `LinkedMonitorDevice` 1:M. Do tejto tabuľky sa ukladajú požiadavky pre jednotlivé prilinkované zariadenia, ktoré slúžia na získanie najaktuálnejšej možnej pozície zariadenia. Po prečítaní požiadavku je požiadavok zmazaný.

6.3.4 Použité knižnice

Microsoft ponúka naozaj širokú ponuku nástrojov, knižníc na tvorbu aplikácií pre Windows Phone. Niektoré funkcie nie sú postačujúce, ale je možnosť použitia knižníc od tretích strán. Knižnice sú pod licenciou, ktorá umožňuje ich voľné používanie a šírenie.

MVVM Light Toolkit

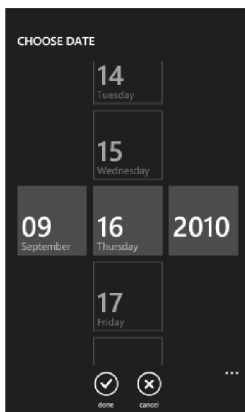
MVVM Light Toolkit¹⁵ knižnica ponúka sadu nástrojov pre vývojárov, ktorí chcú využívať pri tvorbe aplikácií architektonický vzor MVVM¹⁶.

¹⁵ <https://mvvmlight.codeplex.com/>

¹⁶ MVVM - http://en.wikipedia.org/wiki/Model_View_ViewModel

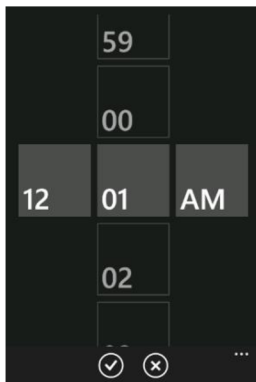
Silverlight Toolkit

Silverlight Toolkit¹⁷ je knižnica, ktorá rozširuje sadu komponent. Knižnica spadá po licenci Microsoft Public License(Ms-PL). Z tejto knižnice je použitý DatePicker, ktorý slúži na vybratie dátumu. Je to veľmi jednoduchá komponenta na ovládanie.



Obrázok 7. DatePicker

Ďalšou komponentou z dielne tejto knižnice, ktorá je použitá je TimePicker. Je to obdoba DatePicker s tým rozdielom, že sa vyberá čas.



Obrázok 8. TimePicker

Poslednou komponentou z tejto knižnice bola komponenta AutoCompleteBox. Funkcionalita AutoCompleteBox je založená na tom, že obsahuje zoznam položiek, buď statický alebo dynamický a pri písaní ponúka užívateľovi jednotlivé položky. Je možné nastaviť aj filter, podľa ktorého sa budú zobrazovať položky, ktoré sa zhodujú s napísaným textom. Táto komponenta je veľmi užitočná pri vyhľadávaní adries, keď užívateľ vytvára profil.

¹⁷ <https://silverlight.codeplex.com/>



Obrázok 9. AutoCompleteBox

BindableApplicationBar

BindableApplicationBar¹⁸ knižnica je obdobou aplikačného panelu, ktorý ponuka Windows Phone Toolkit len s tým rozdielom, že tento panel možno naplno využiť v architektonickom vzore MVVM.

¹⁸ <https://bindableapplicationb.codeplex.com/>

Kapitola 7

Implementácia

Táto časť bakalárskej práce sa zaoberá implementáciou. Kapitola je rozčlenená do viacerých podkapitol, ktoré opisujú jednotlivé kroky postupu implementácie. Aplikácie sú implementované tak, aby boli ľahko udržiavateľné a jednoduchšie rozširiteľné. Vychádza sa zo vzoru MVVM. Tento vzor umožňuje oddelenie časti vizuálnej od časti logickej a dátovej. Implementácia aplikácií prebiehala vo vývojovom prostredí Visual Studio 2010 a databáza bola implementovaná pomocou SQL Server Management Studio.

7.1 Databáza

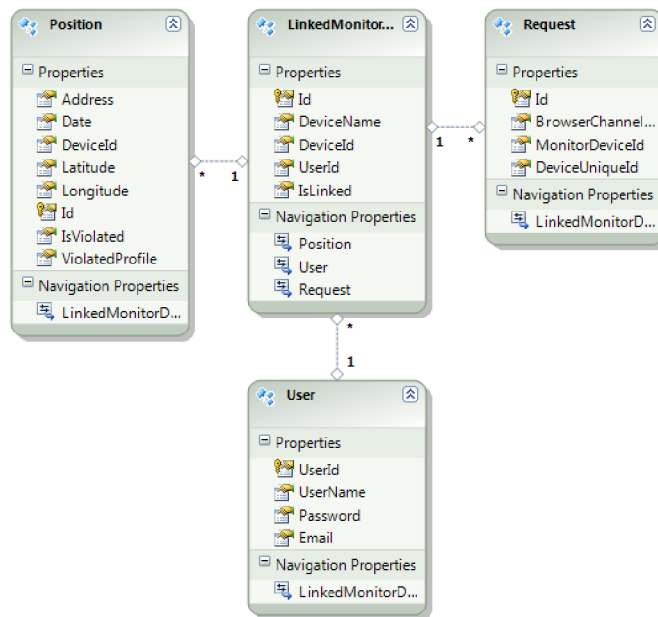
Pri implementácii databázy možno využiť prostredie SQL Server Management Studio, ktoré uľahčuje prácu s databázou a odpadá rúžia písania skriptov SQL. Takto vytvorená databáza slúžila pre účely testovania na lokálnom počítači. Databázu bolo potrebné niekam premigrovať, aby bolo možné aplikáciu testovať aj v reálnom prostredí. Pre tieto účely poslúžil SQL Azure, čo je typ relačnej databázy umiestnenej na Windows Azure. K migrácii na Windows Azure existuje výborný nástroj SQL Database Migration Wizard¹⁹. Tento nástroj vygeneroval SQL skript z lokálnej databázy. Takto vygenerovaný skript sa potom aplikoval na databázu vytvorenú na Windows Azure. Skript povytváral tabuľky a naplnil ich údajmi, ktoré boli obsiahnuté na lokálnej databáze.

7.1.1 Namapovanie databázy

Ako vyplýva z návrhu, s databázou komunikuje WCF servis. K tomu aby mohol servis komunikovať s databázou, bolo nutné databázu namapovať. Visual Studio ponúka rôzne možnosti ako databázu namapovať, napr. LINQ to SQL Classes, ADO.NET Entity Data Model, ai. V aplikácii je použité mapovanie ADO.NET Entity Data Model a to z toho dôvodu, že je viac všestranné, tj. možno takto namapovať aj databázy ako sú MySQL, Oracle, čo v prípade použitia LINQ to SQL Classes nejde, pretože ten dokáže mapovať len MS SQL databázu. Po namapovaní databázy sa

¹⁹ <https://sqlazuremw.codeplex.com/>

vytvorí súbor *.edmx*, ktorý obsahuje diagram na obrázku 10. Namapovanú databázu možno označiť ako *DataContext*. *DataContext* umožňuje prístup do SQL databázy.



Obrázok 10. Namapovaná databáza

Jednotlivé tabuľky predstavujú triedy-typy, ktoré obsahujú vlastnosti jednotlivých stĺpcov. Databázové typy sú mapované na typy .Net, napr. `NAVARCHAR(MAX)` je mapovaný na typ `String`. Celý princíp je založený na tom, že zmeny, ktoré sa vykonávajú za behu aplikácie, tak sa premietajú do *DataContextu*. Z toho vyplýva, že zmeny sa nevykonávajú priamo vo fyzickej databáze. Ak by sme chceli zapísať zmeny priamo do fyzickej databázy, museli by sme zavolať funkciu `SaveChanges()`. Táto metóda vykoná zmeny, či už sa to týka zapísania novej položky do tabuľky, zmazania položky alebo úpravy položky. Pokiaľ chceme nejako narábať s dátami, nepotrebujeme k tomu vytvárať zložité SQL dotazy, ale môžeme využiť triedu LINQ, ktorá výrazne uľahčuje prácu. Na obrázku možno vidieť, ako funguje LINQ a zároveň vidieť aj použitie vyššie zmienenej funkcie `SaveChanges()`.

```
public void UnlinkMonitorDevice(string deviceId)
{
    var entities = new GpsTrackerEntities();
    var device = (from d in entities.LinkedMonitorDevice where d.DeviceId == deviceId select d).FirstOrDefault();
    if(device != null)
    {
        device.IsLinked = false;
        entities.SaveChanges();
    }
}
```

Obrázok 11. Použitie LINQ dotazu

7.2 WCF Servis

Keďže je servis hostovaný na Windows Azure, bolo potrebné vytvoriť špeciálny typ projektu vo Visual Studiu. K tomu, aby bolo možné vytvoriť projekt, ktorý by bolo možné nahrať na Windows Azure, je zapotreby do Visual Studia doinštalovať Windows Azure SDK for .Net. Servis implementuje rozhranie, v ktorom sú deklarované metódy. Metódy, ktoré sú využité na strane klienta tzn. metódy, ktoré sú volané vzdialene musia byť označené atribútom `[OperationContract]`. Celé rozhranie si vyžaduje označenie atribútom `[ServiceContract]`. Na obrázku možno vidieť zjednodušenú implementáciu takéhoto rozhrania.

```
[ServiceContract]
public interface IWcfNotificationService
{
    [OperationContract]
    Status.ServiceStatus LinkMonitorDevice(string userName, string password, string deviceUniqueId, string deviceName);
}
```

Obrázok 12. Implementácia rozhrania servisu

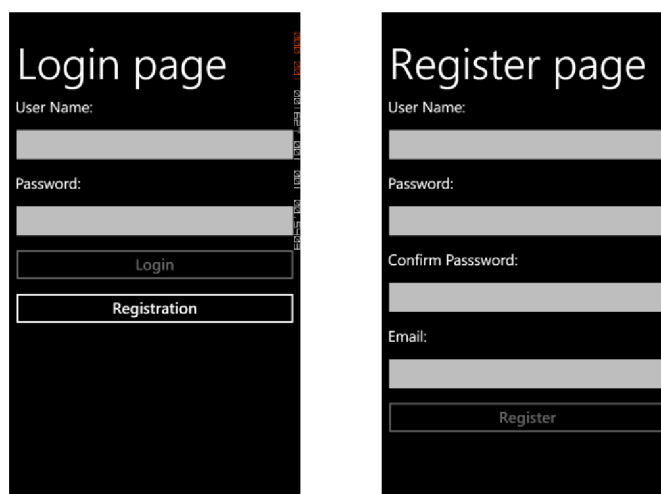
Deklarované metódy je nutné definovať a to tak, že trieda ktorá definuje jednotlivé metódy, musí byť odvodená od rozhrania. Ako vieme, servis komunikuje s databázou. Aby servis vedel, ako sa má pripojiť k databáze, musí poznať tzv. `ConnectionString`. Servis obsahuje nastavenia, ktoré sú uložené v súbore **Web.config**. V tomto súbore sa nachádza aj vyššie spomenutý `ConnectionString`. Ďalšími nastaveniami je napríklad aké má byť vytvorené spojenie medzi servisom a klientom. WCF servis v spojení s Windows Phone umožňuje len spojenie **baseHttpBinding**. Jednou z metód, ktorú obsahuje servis, je posielanie emailov v prípade porušenia časového profilu. Táto metóda si vyžaduje nastavenia ako: SMTP, PORT, NETWORK_CREDENTIALS, HOST. Jednotlivé nastavenia možno nastaviť priamo v kóde, ale z dôvodu predídania nejakým zásahom do kódu a aby šlo o lepšiu udržiavateľnosť, tak sú tieto nastavenia uložené priamo do súboru **Web.config**, z ktorého si ich príslušná metóda načíta. Po implementácii servisu je nutné vytvoriť tzv. **Package**, ktorý obsahuje dva súbory: **ServiceConfiguration.Cloud.cscfg** a **WcfServiceAzure.cspkg**. Tieto dva súbory sa nahrali na Windows Azure, aby mohol byť servis hostovaný. Vzhľadom k tomu, že obaja klienti používajú tento servis, bolo treba zabezpečiť spojenie. O vytvorenie spojenia sa postará Visual Studio, len mu treba predať adresu, na ktorej sa nachádza servis. Tým sa pridá referencia do projektov.

7.3 GpsBrowser

V tejto časti podkapitoly sa budeme zaoberať implementáciou aplikácie GpsBrowser. Podkapitola je rozdelená na časti, v ktorých jednotlivo rozoberieme implementáciu všetkých stránok. Aplikácia GpsBrowser si vyžaduje neustále pripojenie na internet.

7.3.1 Úvodná obrazovka

Po spustení aplikácie uvíta užívateľa úvodná obrazovka, ktorá vyžaduje prihlasovacie údaje, aby bolo možné sa prihlásiť do aplikácie. Ak užívateľ nemá ešte vytvorené konto, tak sa musí registrovať. Po kliknutí na tlačidlo registrácie, dochádza k presmerovaniu na stránku registrácie.



Obrázok 13. Úvodná obrazovka 1 – Prihlasovacia stránka 2 – Stránka registrácie

Pri registrácii je potrebné vyplniť štyri položky: UserName, Password, Confirm Password, Email. Pokiaľ užívateľ nevyplní všetky položky, nemá možnosť kliknúť na tlačidlo Registration. Toto tlačidlo sa aktivuje až v prípade, keď sú všetky položky vyplnené. Táto funkcionlita je zabezpečená pomocou špeciálneho typu `RelayCommand`, ktorý ponúka knižnica `MVVM Light Toolkit`. Po kliknutí tlačidla Registration sa na strane klienta overuje, či sa položka Password zhoduje s položkou Confirm Password. Pri zhode sa následne overuje kritérium, či zadané heslo spĺňa podmienky: minimálne 6 znakov, aspoň jednu číslicu, jedno malé a jedno veľké písmeno a taktiež musí obsahovať aspoň jeden špeciálny znak. Táto syntaktická kontrola sa overuje pomocou regulárneho výrazu - `"((?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%])(?!.*\s){6,})$"`. Takáto syntaktická kontrola sa vykonáva aj pri overovaní položky Email pomocou výrazu - `"^[_A-Za-z0-9-\+\]+\(\.\[_A-Za-z0-9-\+\)*@[A-Za-z0-9-\+\(\.\[_A-Za-z0-9-\+\)*(\.\[A-Za-z0-9-\+\)*\(\.\[A-Za-z]{2,})$"`. Následne po týchto kontrolách sa zavolá funkcia servisu `RegisterUser(User user)`. Parameter funkcie obsahuje všetky údaje, ktoré užívateľ vyplnil,

ale ešte k tomu obsahuje aj jednoznačný identifikátor Guid a heslo v tvare hashu. Hash je vytvorený pomocou triedy `.Net Rfc2898DeriveBytes`, ktorá využíva pseudo-náhodný generátor čísel založený na HMACSHA1. Funkcia overuje, či daný užívateľ už existuje a ak áno, tak užívateľ je vyzvaný zadať nové prihlasovacie údaje. Po úspešnej registrácii dochádza k presmerovaniu na hlavnú stránku aplikácie.

Prihlasovanie do aplikácie zabezpečuje funkcia `VerifyUser(string userName, string Password)`. Funkcia overí, či dané užívateľské meno existuje a zo zadaného hesla vytvorí hash, ktorý porovná s hashom uloženým v databáze, ktorý prislúcha danému užívateľovi. Keď je užívateľ úspešne prihlásený, tak tieto údaje sú uložené aj v nastaveniach aplikácie, pretože sú použité na vytvorenie šifrovacieho kľúča, ktorý slúži k dešifrovaniu údajov uložených v databáze.

7.3.2 Hlavná obrazovka

Hlavná obrazovka obsahuje len jedno tlačidlo, ktoré slúži na zobrazenie zariadení, ktoré má užívateľ prilinkované.

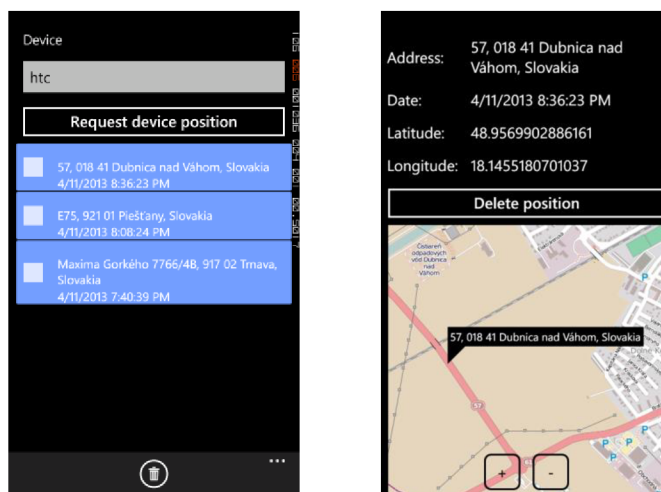


Obrázok 14. Hlavná obrazovka

7.3.3 Zoznam zariadení

Nasledujúcou stránkou je stránka zoznamu zariadení. Pri prechode na túto stránku sa načítajú všetky zariadenia, ktoré užívateľ prilinkoval. Ak nie sú k dispozícii žiadne zariadenia, tak sa zobrazí informácia o tom, že neboli nájdené žiadne zariadenia. Načítanie zariadení má na starosti funkcia `GetLinkedMonitorDevices(string userName, byte[] key)`. Funkcia vyhledá zariadenia, ktoré prislúchajú užívateľovi vrátane pozícií, ktoré majú zariadenia uložené. Pozície sú dešifrované pomocou kľúča. Dešifrovanie sa vykonáva na strane servera pomocou funkcie `DecryptData(byte[] encryptedData, byte[] key)`. Ako výsledok funkcie `GetLinkedMonitorDevices` je vrátená kolekcia všetkých zariadení.

Na zobrazenie zariadení je použitá komponenta ListPicker. Pri vybratí nejakého zariadenia sa zobrazí kolekcia všetkých jeho pozícií. Ak je označené zariadenie, je možné poslať žiadosť o zistení jeho polohy. Túto funkcionality vykonáva funkcia `SendRequestToMonitorDevice(Guid id, string deviceUniqueId)`. Pozície sú uložené v kolekcii typu `ObservableCollection`, ktorá je pripojená na komponentu `ListBox`. Výhodou tejto kolekcie je to, že ak sa z nej odstráni položka, automaticky sa to prejaví aj v komponente `ListBox`. `ListBox` obsahuje `ListBoxItems`, kde jeden `ListBoxItem` predstavuje jednu pozíciu. `Silverlight` dáva možnosť a voľnosť upraviť si vzhľad `ListBoxItem` podľa vlastných potrieb. `ListBoxItem` vyzerá tak, že sa v ňom zobrazuje `CheckBox`, dátum a adresa pozície. `CheckBox` slúži na označenie položiek, ktoré majú byť zmazané. Na zmazanie pozícií je vytvorené tlačidlo umiestnené v `ApplicationBar`. `ListBox` má v sebe zabudovanú udalosť `OnTap`, ktorá sa vyvolá, keď sa klikne na nejakú položku v `ListBoxe`. V tomto prípade sa pri vyvolaní tejto udalosti prejde na novú stránku, ktorá zobrazuje detailnejší popis pozície. Táto stránka obsahuje adresu, dátum a súradnice pozície. Okrem toho je tu aj tlačidlo na zmazanie pozície a mapa, na ktorej sa zobrazí bod, podľa súradníc. Implementácia mapy je vysvetlená neskôr.



Obrázok 15. 1 – Zoznam zariadení 2 – Detailné zobrazenie pozície

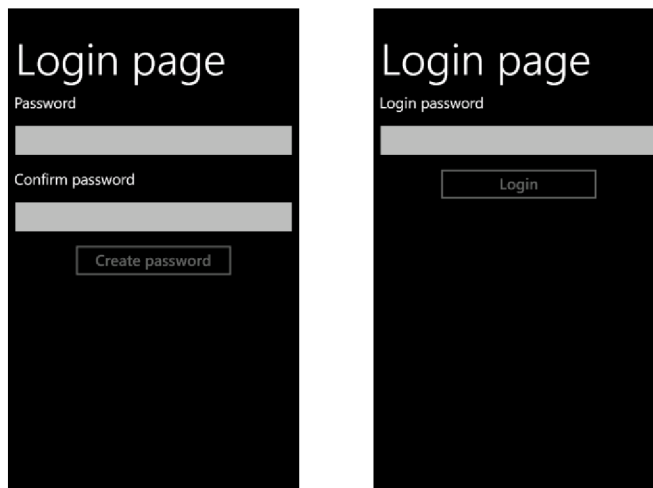
7.4 GpsMonitor

Táto časť podkapitoly sa venuje implementácii aplikácie GpsMonitor.

7.4.1 Úvodná obrazovka

Ako aj v prípade aplikácie GpsBrowser, tak aj v aplikácii GpsMonitor je potrebné prihlasovanie, len s tým rozdielom, že do aplikácie sa neprihlasuje pomocou údajov, ktoré sa využívajú v prípade GpsBrowser. Pri úplne prvom spustení aplikácie sa zobrazí stránka, ktorá požaduje vytvorenie hesla. Heslo musí spĺňať dĺžku minimálne 4 znaky. Vytvorené heslo je uložené v nastaveniach aplikácie a opäť je z bezpečnostných dôvodov uložený jeho hash. Pri ďalšom spustení

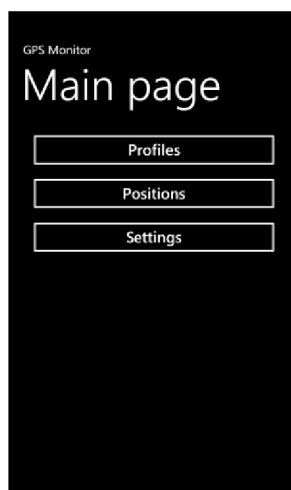
aplikácie je potrebné už len zadať heslo. Toto opatrenie má zabezpečiť to, aby nikto nemohol meniť nastavenia aplikácie apod. Okrem hesla sa do nastavení aplikácie ukladá aj unikátne číslo zariadenia, ktoré sa získa pomocou funkcie `GetDeviceUniqueId()`.



Obrázok 16. 1 – Vytvorenie hesla 2 – Prihlásenie

7.4.2 Hlavná obrazovka

Po tom ako je užívateľ úspešne prihlásený, je presmerovaný na hlavnú stránku. Hlavná stránka obsahuje tri tlačidlá: Profiles, Positions, Settings. Tlačidlo Profiles presmerováva na stránku s vytvorenými profilmi, tlačidlo Positions zobrazí stránku s uloženými pozíciami a nakoniec tlačidlo Settings zobrazuje nastavenia aplikácie.



Obrázok 17. Hlavná obrazovka

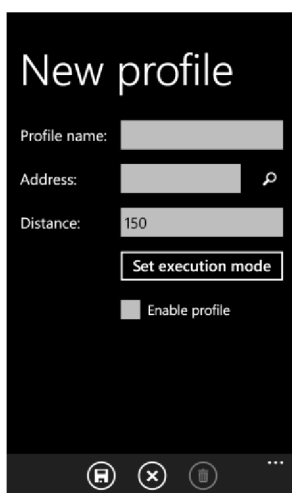
7.4.3 Obrazovka profilov

Obrazovka profilov slúži na vytváranie profilov. Obsahuje jeden ListBox, na ktorý je pripojená kolekcia ObservableCollection s vytvorenými profilmi. Po tom, ako je stránka zobrazená, sa pomocou funkcie ReadFromXml<T>(string filePath) načítajú všetky vytvorené profily zo súboru **profiles.xml** do kolekcie. Ako možno vidieť funkcia vracia generický typ, tak je možné túto funkciu využiť s akýmkoľvek typom. Stránka obsahuje aj ApplicationBar, na ktorom sú umiestnené tlačidlá pre pridanie profilu a zmazanie profilu.

7.4.4 Pridanie profilu

Pridanie profilu pozostáva z viacerých krokov a je rozdelené na viac stránok. Je to z dôvodu prehľadnosti. Naskytuje sa možnosť použiť stránky typu Panorama, ale tá sa nepoužíva na účely typu wizard, čo vytvorenie profilu je.

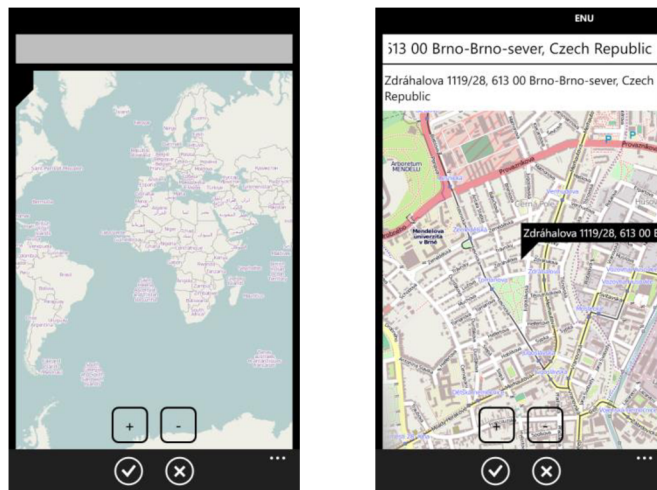
Prvá stránka obsahuje pole pre názov profilu, pole pre adresu, pole pre vzdialenosť – musí byť minimálne 150m, CheckBox pre povolenie profilu a tlačidlo pre nastavenie módu spúšťania. V ApplicationBar sú umiestnené tlačidlá na uloženie, vymazanie profilu a tlačidlo pre zrušenie.



Obrázok 18. Hlavná stránka nastavenia profilu

Do poľa pre adresu sa adresa nevpisuje, ale je vedľa neho umiestnené tlačidlo lupy. Toto tlačidlo zobrazí novú stránku s mapou. Okrem mapy je tu aj komponenta AutoCompleteBox. Na implementáciu mapy je použitá komponenta Map. Komponenta využíva Bing Maps a k svojmu behu vyžaduje ApplicationId, ktoré možno získať registráciou na stránkach Microsoftu. Výhodou Map komponenty je to, že možno na ňu aplikovať pomocou MapTileLayer vlastné zobrazenie mapy(OpenStreetMap, Nokia maps). V aplikácii je použité zobrazenie prostredníctvom OpenStreetMap a to z toho dôvodu, že poskytuje detailnejšie zobrazenie ako Bing Maps.

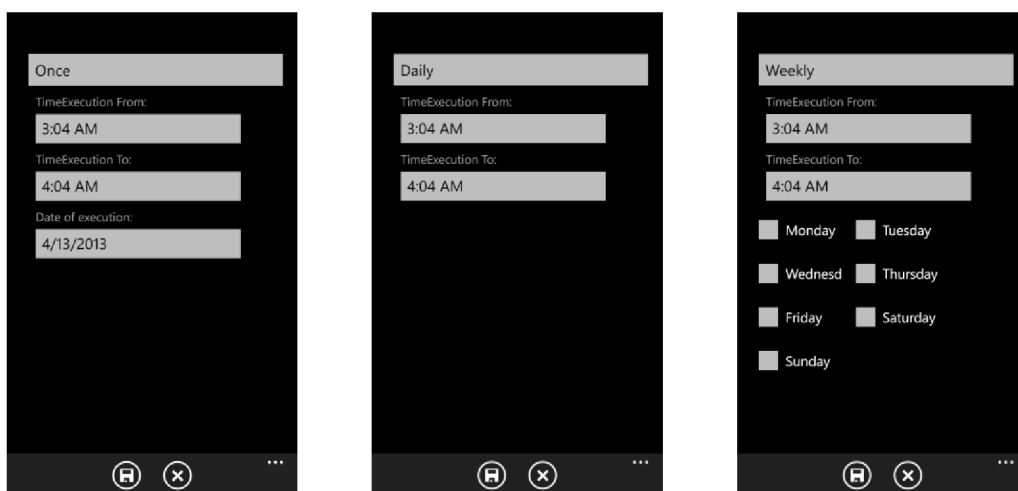
Užívateľ má dve možnosti ako vybrať adresu. Prvou možnosťou je, že do komponenty `AutoCompleteBox` napíše adresu. Minimálne je treba zadať 5 znakov. Vyhľadávanie adries funguje na princípe ako je to u vyhľadávania adries v Google Maps. Využitie je k tomu Google Maps Api. Je to implementované vo vlastnej knižnici, ktorá má dve funkcie: `GeocodingAsync(string address)` a `ReverseGeocodingAsync(double latitude, double longitude)`. Vyhľadávanie adries pomocou `AutoCompleteBox` zabezpečuje prvá funkcia. Vstupný parameter jej slúži adresa, ktorú užívateľ zadal. Funkcia vytvorí požiadavku a odošle ho. Ako odpoveď príde textový výstup s výsledkami zodpovedajúcimi zadanej adrese. Response ma výstupný formát JSON. Aby bolo možné rozparsovať odpoveď, je zapotreby si vytvoriť vlastné typy, ktoré odpovedajú danému formátu výstupu. K rozparsovaniu odpovede je použitá knižnica `Newtonsoft.Json`. Nájdene výsledky sú ponúknuté užívateľovi. Maximálne sa zobrazujú 3 výsledky. Po vybratí adresy sa na mape zobrazí bod s konkrétnou adresou.



Obrázok 19. 1 – Počiatočný stav 2 – Vyhľadanie adresy

Druhou možnosťou, ako vybrať adresu, je podržaním prstu niekde na mape. Vyvolá sa udalosť `OnHold`, ktorá zavolá metódu `ReverseGeocodingAsync(double latitude, double longitude)`. Princíp získania adresy je rovnaký ako v prvom prípade, rozdiel je len v tom, že adresa sa získava zo súradníc. Súradnice sa vypočítajú pomocou funkcie `ViewportPointToLocation(Point p)`. Ide o funkciu, ktorú implementuje komponenta `Map`.

K tomu aby bol profil kompletný treba ešte vytvoriť mód, kedy je daný profil platný. Sú tri možnosti módu: Once, Daily, Weekly.



Obrázok 20. 1 – Once 2 – Daily 3 – Weekly

Na výber módu je použitá komponenta `ListPicker`, na ktorú je pripojený výčtový typ `TypeMode`. K výberu dátumu a času sú použité komponenty `TimePicker` a `DatePicker`. Dátum musí byť zhodný s aktuálnym dátumom alebo musí byť väčší. Nemožno zadať dátum, ktorý sa týka minulosti. V prípade výberu času, musí byť čas `Od` menší ako čas `Do` a zároveň čas `Do` musí byť o 1 hodinu väčší.

7.4.5 Obrazovka pozícií

Obrazovka pozícií zobrazuje pozície, ktoré neboli nahraté na server. Ide o tzv. režim bez použitia internetového pripojenia. Tento režim nastáva vtedy, ak užívateľ nevytvoril link pre dané zariadenie alebo zariadenie nemá prístup na internet. Pozície sa ukladajú do súboru **positions.xml**. Ukladanie pozícií zabezpečuje funkcia `WriteToXml<T>(T data, string filePath, FileMode mode)`. Je tu možnosť jednotlivé pozície nahráť na server alebo je možnosť aj pozície vymazať. Ako v prípade aplikácie `GpsBrowser`, tak aj tu sa dá zobrazit' detailný náhľad na pozíciu.

7.4.6 Obrazovka nastavení

Táto obrazovka obsahuje položku `Email`, kde užívateľ musí zadať emailovú adresu, ktorá slúži na zaslanie notifikácie ak dôjde k porušeniu profilu. Ďalej sa tu nachádza `CheckBox`, ktorý má informatívny charakter o tom, či je zariadenie prilinkované. K dispozícii má užívateľ dve tlačidlá – `Link` a `Unlink`, z ktorých je vždy práve jedno povolené. Ak zariadenie nie je prilinkované, tak je povolené tlačidlo `Link`. Po stlačení tohto tlačidla je užívateľ presmerovaný na stránku, na ktorej zadá prihlasovacie údaje a meno zariadenia.

O vytvorenie linku sa stará funkcia `LinkMonitorDevice(string userName, string deviceUniqueId, string deviceName)`. Funkcia overí, či sú platné prihlasovacie údaje a či dané meno zariadenia pre dotyčného užívateľa náhodou neexistuje. V prípade úspešného prilinkovania sa vytvorí záznam v tabuľke `LinkedMonitorDevice`. Pri odstránení odkazu nie je dané zariadenie z tabuľky odstránené, len mu je nastavený príznak `IsLinked=False`.

Poslednou a najdôležitejšou časťou nastavení je zapnutie monitorovania polohy. K tomuto účelu slúži komponenta `ToggleSwitch`. Zapnutie/Vypnutie značí, či sa pri uložení nastavení vytvorí/odstráni nová úloha v systéme. Z hľadiska prehľadnosti je vytvorená táto úloha ako samostatný projekt podľa šablóny `Visual Studio`. Úloha je do systému pridaná pomocou `ScheduledActionService.Add(task)`. Každých 30 minút sa úloha spúšťa. Pri spustení sa vyvolá metóda `OnInvoke(ScheduledTask task)`, ktorá spustí vyhodnocovaciu logiku. Vyhodnocovacia logika je implementovaná v triede `EvaluationLogic` s jednou verejnou metódou `Evaluate()`. Pri vyhodnocovaní sa získa pozícia pomocou triedy `GeoCoordinateWatcher`. Po získaní pozície sa vyhľadajú všetky platné profily. Funkcia `CalculateDistance(GeoCoordinate actualCoordinate, GeoCoordinate previousCoordinate)` obsahuje implementáciu Haversine formuly, ktorá vypočíta vzdialenosť medzi adresou v profile a získanou pozíciou. Návrátová hodnota sa porovná so vzdialenosťou určenou v profile a podľa toho sa rozhodne, či ide alebo nejde o porušenie profilu. Ak bol profil porušený, tak pozícii sa nastaví príznak o tom, že bola porušená a meno porušeného profilu. Nová pozícia sa aktualizuje po každých 100m, aby nevznikali duplicitné pozície pokiaľ užívateľ stojí aj niekoľko hodín na mieste. V prípade, že pozícia sa neaktualizovala a zariadenie je prilinkované, je potrebné skontrolovať, či niekto nepožaduje od zariadenia pozíciu. O toto sa stará funkcia `SendDataWithoutSaving(string email, string deviceUniqueId, Position position)`, ktorá ale neukladá pozíciu na server. `SendDataServer(string email, string deviceUniqueId, string deviceName, byte[] key, Position position)` je funkciou, ktorá sa zavolá, ak sa pozícia aktualizovala a zariadenie je prilinkované. Zabezpečuje uloženie pozície do databázy, zaslanie emailu ak došlo k porušeniu profilu a odosiela email aj v prípade žiadosti o pozíciu. V prípade, že pozícia sa aktualizovala, ale zariadenie nemá vytvorený odkaz alebo nemá k dispozícii internetové pripojenie, tak je pozícia uložená lokálne pomocou funkcie `SavePosInFile(Position position)`. Operácie ako načítanie alebo zápis na úložisko telefónu museli byť synchronizované, lebo by mohlo dôjsť k problému, ak by hlavná aplikácia čítala/zapisovala dáta a v tom istom čase by sa spustila aj úloha na pozadí, ktorá číta/zapisuje dáta, tak by nastalo uviaznutie. Synchronizácia je vyriešená pomocou mutexov.



Obrázok 21. Nastavenia

Kapitola 8

Testovanie

Implementácia prebiehala z väčšej časti na emulátore. Emulátor ale nebol schopný poskytnúť reálne možnosti pre testovanie. Z toho dôvodu som si zapožičal mobilný telefón od školy. Išlo o telefón HTC Titan s Windows Phone 7.5. Aby bolo možné testovať aplikácie, musel som telefón zaregistrovať. Po úspešnej registrácii som mohol nahráť aplikácie do telefónu. Aplikácie sa spustili na prvý krát bez akýchkoľvek problémov. Nahráť samotné aplikácie na telefón nestačilo. Aby sa dalo testovať aplikáciu reálne, museli byť databáza a aj servis niekde hostovaný. Pri použití emulátoru stačilo mať servis a aj databázu hostované na lokálnom počítači. Po zvážení všetkých možností som sa rozhodol využiť Windows Azure, ktorý je na 90 dní zadarmo. Keďže nie všade je pripojenie na Wi-fi, aktivoval som si internet u mobilného operátora. Po dobu 14 dní boli aplikácie v prevádzke. Obe aplikácie fungovali podľa očakávaní. Aplikácie moc telefón nevyťažovali, takže sa to neprejavilo ani na veľkej spotrebe batérie. Najväčšia spotreba nastávala len vtedy, keď sa spúšťal proces na získanie pozície, pretože práca s GPS modulom má najväčšiu režiu na spotrebu. Internet, ktorý bol zaobstaraný u mobilného poskytovateľa, úplne postačuje na tento typ aplikácie. Konkrétne som mal k dispozícii internet s kapacitou dát 100MB. Po dvojtýždňovom testovaní bol počet prenesených dát okolo 5MB, čo nepredstavuje až také veľké číslo. Aplikácia fungovala za každého počasia a na každom mieste. Pokiaľ sa nachádzal sledovaný objekt v nejakej budove, tak bola zaznamenaná približná poloha zariadenia.

Kapitola 9

Záver

Cieľom tejto práce bolo vytvoriť primárne aplikáciu na sledovanie polohy zariadenia a profilovanie jednotlivých polôh. Okrem iného bolo potrebné zaznamenávať rôzne činnosti, ktoré sa vykonávali na telefóne, ale z hľadiska bezpečnostného nešlo túto funkcionality implementovať. Vytvoril som dve aplikácie GpsMonitor pre zaznamenávanie polohy, vytváranie profilov, vyhodnocovanie profilov a GpsBrowser pre prehliadanie polôh umiestnených na servery. Pre tento druh aplikácií pod Windows Phone som sa rozhodol preto, lebo som mal už skúsenosti s vývojom v jazyku C# a s prácou vo vývojom prostredí Visual Studio 2010. Aplikáciu hodnotím ako stredne náročnú.

Možné rozšírenie aplikácie vidím v použití nových vlastností, ktoré sú už dostupné vo Windows Phone 8, ako napríklad získanie reálnej polohy v procese bežiacom na pozadí, pretože Windows Phone 7.5 umožňuje získať len pozíciu, ktorá sa aktualizuje každých 15 minút. Ďalšie rozšírenie by som videl v možnosti meniť šifrovacie kľúče. Ak to niekedy umožní API, tak rozšíriť zasielanie notifikácií prostredníctvom sms správ. V budúcnosti by som chcel pokračovať na aplikácii, plánujem ju zverejniť na Marketplace.

Využitie samotnej aplikácie je pomerne úzke. Aplikácia je primárne určená pre sledovanie osôb, hlavne detí a zamestnancov, takže môže slúžiť ako pomocník pre rodičov, aby mohli sledovať, kde sa nachádzajú ich deti, napr. počas obdobia školy alebo to môže poslúžiť pre zamestnávateľa ako nástroj pre sledovanie zamestnanca počas služobnej cesty.

Literatura

- [1] BABCANÍK, Jan. *Jak funguje GPS?* [online]. 23.6. 2006 [cit. 2013-04-14]. Dostupné z: <http://hw.cz/Teorie-a-praxe/ART1634-Jak-funguje-GPS.html>
- [2] Wikipedia. *Haversine formula* [online]. 27.2. 2013 [cit. 2013-04-14]. Dostupné z: http://en.wikipedia.org/wiki/Haversine_formula
- [3] Wikipedia. *Windows Phone 7* [online]. 23.3. 2013 [cit. 2013-04-14]. Dostupné z: http://en.wikipedia.org/wiki/Windows_Phone_7
- [4] PETZOLD, Charles. *Programming Windows Phone 7*. Washington : Microsoft Press, 2010. 997 s. ISBN 978-0-7356-4335-2.
- [5] VAUGHAN, D. *Windows Phone 7.5 Unleashed*. Sams, 2012. 1095 s. ISBN 978-0-672-33348-4
- [6] Wikipedia. *Microsoft Silverlight* [online]. 12.4. 2013 [cit. 2013-04-14]. Dostupné z: http://en.wikipedia.org/wiki/Microsoft_Silverlight
- [7] MSDN. *What Is Windows Communication Foundation* [online]. 2.8. 2012 [cit. 2013-04-14]. Dostupné z: <http://msdn.microsoft.com/en-us/library/ms731082.aspx>
- [8] KOŠŤÁL, Marián. *Začínáme s WCF* [online]. 1. Február 2007 [cit. 2013-04-14]. Dostupné z: <http://www.vyvojar.cz/Articles/452-zaciname-s-wcf.aspx>
- [9] NAGEL , Christian, et al. *C# 2008 : Programujeme profesionálně. [s.l.]* : Computer press, 2009. 1904 s. ISBN 978-80-251-2401-7.
- [10] ZVÁNOVEC, Vít, et al. *Ochrana osobních údajů nezletilců* [online]. 30.12. 2008 [cit. 2013-04-14]. Dostupné z: <http://www.pravnickeforum.cz/archiv/dokument/doc-d36735v46624-ochrana-osobnich-udaju-nezletilcu/>
- [11] JAKUBKA, Jaroslav. *Ochrana osobních a osobnostních práv zaměstnanců – II. část* [online]. 5.12. 2005 [cit. 2013-04-14]. Dostupné z: <http://www.mzdovapraxe.cz/archiv/dokument/doc-d1073v1049-ochrana-osobnich-a-osobnostnich-prav-zamestnancu-ii-cast/>

Zoznam príloh

Príloha A. Obsah CD

Príloha A. Obsah CD

Priložené CD obsahuje tieto súčasti:

- Zdrojové kódy – obsahuje projekt pre Visual Studio 2010 vrátane potrebných knižníc
- Spustiteľný program – obsahuje binárne súbory
- Text – obsahuje PDF súbor s textom bakalárskej práce
- Manuál – obsahuje súbor s manuálom