



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VÍCEUŽIVATELSKÁ MAPOVÁ APLIKACE PRO MOBILNÍ ZAŘÍZENÍ

MULTIUSER MAPPING APPLICATION FOR MOBILE DEVICE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN UTĚKAL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JOZEF MLÍCH

BRNO 2012

Abstrakt

Tato diplomová práce pojednává o návrhu a implementaci víceuživatelské mapové aplikace pro mobilní zařízení. Cílem práce je vytvoření koordinačního systému určeného primárně pro řízení mobilních jednotek Policie České republiky. Nejprve jsou popsány již existující řešení, které s aplikací sdílejí podobné zaměření. Následně jsou představeny různé komunikační, mapové a databázové technologie, z nichž jsou vybrány ty, které se pro účely práce hodí nejlépe. Další částí práce je popis návrhu systému a jeho implementace. Efektivitu aplikace v praxi ověřuje série testů, které se zabývají energetickou a datovou náročností aplikace. Práci uzavírá představení možných budoucích rozšíření programu a směry jeho dalšího vývoje.

Abstract

This thesis presents design and implementation of multiuser mapping application for mobile device. The aim is to create a coordination system intended primarily for controlling mobile units of the Police of the Czech Republic. First, the existing solutions with similar focus are described. Subsequently, the most suitable communication, mapping and database technologies are selected. The next part of this work describes the system design and its implementation. Effectiveness of the system is verified in a series of tests, which deal with the power and data consumption. The thesis concludes with possible extensions of the program and ways of future development.

Klíčová slova

Mapa, navigace, Maemo, MeeGo, Symbian, XMPP, SVG, PostgreSQL, Qt

Keywords

Map, navigation, Maemo, MeeGo, Symbian, XMPP, SVG, PostgreSQL, Qt

Citace

Jan Utěkal: Víceuživatelská mapová aplikace pro mobilní zařízení, diplomová práce, Brno, FIT VUT v Brně, 2012

Víceuživatelská mapová aplikace pro mobilní zařízení

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana
Ing. Jozefa Mlícha

.....
Jan Utěkal
21. května 2012

Poděkování

Tímto děkuji vedoucímu své diplomové práce Ing. Jozefovi Mlíchovi za jeho vedení, rady,
nápady a inspirace, které mi v průběhu vypracování práce nabídl.

© Jan Utěkal, 2012.

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

Obsah

1	Úvod	2
2	Existující řešení	4
2.1	Koordinační systém Policie ČR	4
2.2	GINA	5
2.3	Ostatní	6
3	Mobilní platformy	8
3.1	Maemo a MeeGo	8
3.2	Symbian	10
3.3	Qt Mobility	11
4	Technologie	13
4.1	Mapové podklady	13
4.2	Zobrazení mapy	14
4.3	Grafická reprezentace dat	15
4.4	Databázové systémy	16
4.5	Komunikace	18
4.6	XMPP	21
5	Návrh	24
5.1	Případy užití	24
5.2	Architektura	25
6	Implementace	29
6.1	Funkcionalita	29
6.2	Popis uživatelského rozhraní	36
6.3	Problémy při implementaci	37
7	Testování a další vývoj	40
7.1	Výdrž baterie a datové nároky	40
7.2	Rozšíření a možná nasazení aplikace	42
8	Závěr	44
A	Obsah DVD	48

Kapitola 1

Úvod

Žijeme v době, kdy jsou informace jednou z nejdůležitějších součástí našich životů. Dnes už si jen stěží dokážeme představit, že bychom neměli nejnovější zprávy ze světa několikrát za den na očích, kdy bychom netušili, jak se mají naši známí, vzdálení od nás fyzicky stovky kilometrů, nebo kdy bychom se museli dívat do mapy, abychom se dostali na nějaké konkrétní místo.

Moderní technologie se staly tak běžnou a všední skutečností, že lze jen těžko očekávat jiný vývoj, než jejich stále větší integraci do našeho života. Trend jasně říká, že kdo dnes není na internetu, ten jako by ani nebyl a tak se bez internetu neobejdou naše počítače, ale ani telefony, televize nebo auta. Tato situace vede k tomu, že to, co dříve mohlo vyžadovat kupříkladu dlouhé studium v knihovně, lze dnes získat s pomocí mobilního telefonu za zlomek času.

S rozvojem mobilních zařízení rostou možnosti člověka, jak být vždy na příjmu, jak získávat informace ne jen tehdy, když si sedne ke svému počítači, ale vždy když to chce, nebo kdy to situace vyžaduje. Prostředkem k tomu jsou velmi často mobilní aplikace, které různou kombinací vlastností telefonu a jeho okolí dovolují uživateli pomocí několika málo pohybů dosáhnout svých cílů. Vznikají tak různá řešení s mobilním telefonem jako klientem a serverem na internetu, která umožňují uživateli ve větší, či menší míře dělat za pohybu to, na co dříve potřeboval stolní počítač.

Práce, kterou držíte v ruce, je jedním takovým provázáním současných, zejména geolokačních a komunikačních, technologií do jednoho celku. Ten si klade za cíl zvýšit efektivitu práce jednotlivců nebo skupin pohybujících se v terénu. To realizuje tak, že zaznamenáváním jejich polohy dovoluje přesně plánovat trasy jejich cest, nebo vytyčovat oblasti, ve kterých se mají pohybovat. Ačkoli usiluje o platformní a aplikační nezávislost, volí si práce konkrétní modelový případ, pro který implementuje řešení. Tímto případem je navigační systém pro Policii ČR. Ta v současné době disponuje systémem pro řízení svých jednotek. Tento systém má však výrazná omezení a může sloužit výhradně motorizovaným jednotkám. Z toho důvodu si tato práce klade za cíl vytvořit univerzální koordinační systém, který umožňuje řídit jak jednotky motorizované, tak i pěší. Kvůli důrazu na univerzálnost řešení však může vytvořený systém po jistých modifikacích sloužit kupříkladu soukromým firmám.

Kapitola 2 představuje existující softwarová řešení, která se svým zaměřením představované aplikace podobají. U každého z nich jsou zmíněny společné rysy a jsou představeny aspekty, v nichž se implementovaný systém liší.

V následující kapitole 3 je představena cílová platforma Maemo a jsou zde popsány i další platformy, MeeGo a Symbian, o které bylo zadání práce rozšířeno. U všech jsou popsány specifické prvky a nástroje, kterými se odlišuje vývoj pro ně od vývoje na jiné

platformy.

Ve 4. kapitole jsou diskutovány jednotlivé technologie, které je možné v rámci této práce použít. U každé z nich jsou zmíněny požadavky, které na ni práce klade a v tomto kontextu je následně zhodnocen její potenciální přínos. Představeni jsou poskytovatelé mapových služeb a komponenty, které jsou schopny je zobrazovat. Diskutovány jsou také způsoby, jak reprezentovat posílaná data a jakým způsobem je perzistentně ukládat. V závěru kapitoly jsou zhodnoceny komunikační protokoly, které je možné v systému využít, a je blíže představen protokol XMPP, který se pro účely práce hodí nejlépe.

V kapitole 5 je představen návrh systému. Probrány jsou jak případy užití, tak i architektura systému. Následující kapitola 6 se věnuje způsobu, jakým byl navržený systém implementován. Představena je jak funkcionalita, tak i uživatelské rozhraní a problémy, které při vývoji vyvstaly.

Poslední, 7. kapitola se zabývá testováním implementovaného systému z hlediska spotřeby energie na mobilních zařízeních a nároků na datové připojení. V další části pak diskutuje jeho další vývoj a případné nasazení.

Kapitola 2

Existující řešení

Existuje několik softwarových řešení, které jsou představovanému systému v některých ohledech podobné. Tematicky nejbližší je již existující koordinační systém Policie České republiky. Druhým je aplikace GINA společnosti GINA Software¹. Tyto dva systémy jsou v rámci této kapitoly blíže představeny a porovnány s navrhovaným řešením z hlediska cíle jejich nasazení a způsobu, jakým tento cíl plní. V závěru kapitoly jsou dále zmíněny některé další systémy, které s navrhovaným systémem sdílí společné prvky, ale jejichž zaměření je odlišné. Tato kapitola vychází z [15].

2.1 Koordinační systém Policie ČR

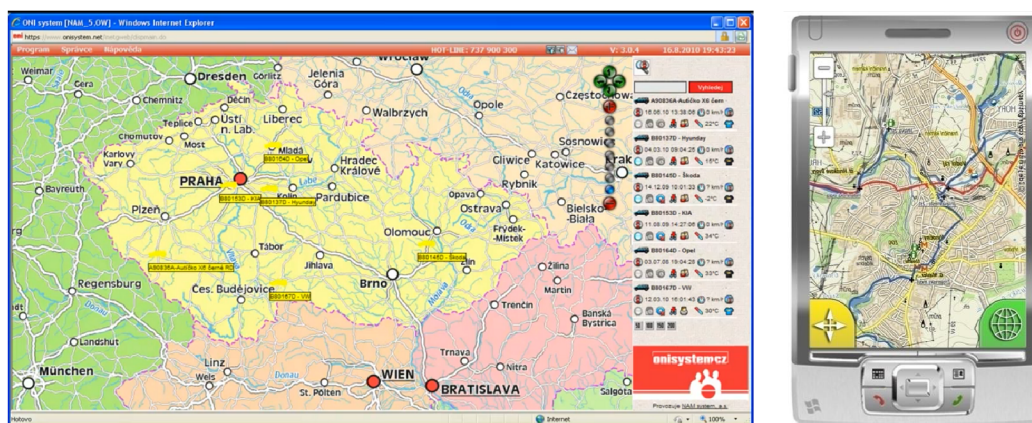
Policie ČR v současné době již disponuje řešením, které se z hlediska svého zaměření navrhovanému systému podobá. Na obrázku 2.1 je vlevo vidět jeho uživatelské rozhraní. Cílem programu je koordinace motorizovaných hlídek ve městech. Ta je realizována prostřednictvím dispečera, který má k dispozici seznam hlídek, pro něž může označit místo, na které se mají vydat. Pro toto místo může zadat popis s délkou 256 znaků. Hlídky mají ve svém automobilu GPS zařízení, které zachycuje jejich polohu a tu odesílá na centrálu. Kromě toho mají také standardní zařízení pro navigaci s mapou, na níž se zobrazuje obdržení bod i s popisem, po jehož označení začne software hlídku navigovat.

Systém využívá mapové podklady od společnosti Mapy.cz, které může zobrazovat jak v klasické formě, tak i v režimu satelitní mapy. K zobrazení mapy využívá software webového klienta, v němž je možné taktéž analyzovat a zobrazovat trasy vozidel v určitém časovém rozmezí. Snímač GPS polohy je vybaven zařízením pro zaznamenání přetížení, kterým lze odhalit havárii vozidla. V případě že tento snímač zachytí přetížení přesahující hodnotu 3g, je na centrálu zaslán poplašný signál a ta může podle polohy vozidla jednat.

Nevýhodou tohoto systému je, že je pevně zabudován do palubní desky automobilu, a tudíž je využitelný pouze pro motorizované jednotky. Úprava automobilu navíc není triviální a je i poměrně nákladná, což v kombinaci s měsíčními poplatky za každé vestavené zařízení zvyšuje celkovou cenu systému. Druhou nevýhodou je, že je možné v mapě označit pouze jeden konkrétní bod, k němuž lze přiřadit krátký popis. V případě, že během pohybu jednotky dispečer zjistí další informace, nemůže je již jednotce v rámci systému dodatečně předat. Navíc je silně omezen nemožností vytvářet na mapě dodatečné grafické objekty a software tudíž může sloužit pouze pro vzdálené zadávání míst, kam se má dopravit některá z motorizovaných jednotek, jíž Policie disponuje.

¹<http://www.ginasoftware.cz/>

Z výše uvedené specifikace vyplývá, že je stávající systém určen výhradně pro řízení hlídek, které musí být pro užívání systému fyzicky přítomny uvnitř automobilu. Pro případy, kdy je takováto obsluha dostačující, je systém vhodný a plní svůj účel. V okamžiku, kdy ale jednotka automobil opustí, nemůže již dále výhod systému využívat a musí se spolehnout na standardní postup. Aplikace, kterou tato práce představuje, využívá jako mobilní zařízení telefonu, který tak nabízí širší možnosti, kdy může plnit všechny zmíněné funkce, pokud se jednotka nachází uvnitř automobilu, ale zároveň nabízí široké možnosti v momentě, kdy jednotka vozidlo opustí. Pro jeho užívání navíc není nutné, aby byla jednotka motorizovaná a navrhovaný systém tak může sloužit stejně dobře pěším hlídkám, které s sebou mobilní telefon i tak nosí. Díky možnosti zakreslovat do mapy komplexnější grafické objekty, která bude představena dále v této práci, je jej navíc možné využívat i pro složitější činnosti, jako je koordinace policistů při prohledávání dané oblasti a podobně.



Obrázek 2.1: Systém policie ČR a GINA

2.2 GINA

Druhým podobným, již existujícím, řešením je aplikace GINA, kterou lze vidět na obrázku 2.1 vpravo. Jedná se o systém, jehož hlavním cílem je komunikace uživatelů, kteří se pohybují v terénu, pomocí zaznamenávání jejich geografické pozice zakreslováním obrázků do mapy. Mapa se nachází na mobilním telefonu s operačním systémem Windows Mobile. Aplikace využívá mapové podklady OpenStreetMap ve čtyřech variantách.

Hlavním rozdílem jak proti navrhovanému systému, tak i proti tomu, který využívá Policie ČR je, že GINA nemá stacionárního koordinátora, který by se staral o řízení ostatních uživatelů, ale je založena na výměně informací mezi jednotlivými uživateli. Toho dosahuje tak, že implementací vlastní webové služby sbírá data od všech uživatelů a ty následně rozepisuje těm uživatelům, kterým jsou tato data určena. Kterým uživatelům data náleží určuje tzv. session, což je v podstatě skupina, do níž se uživatelé ručně přihlašují. Přihlášením se k session tedy uživatel určuje, komu bude odesílat data a jaká data se mu budou zobrazovat.

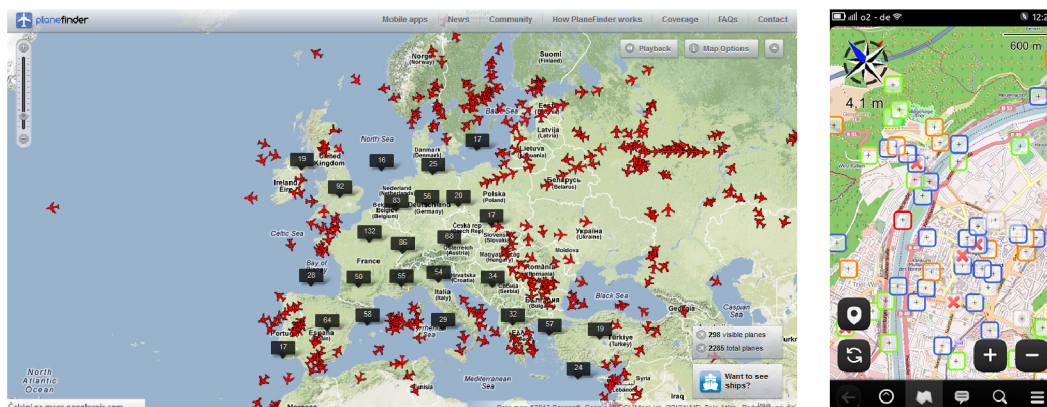
Aplikace má sloužit zejména v krizových situacích, jako bylo zemětřesení na Haiti v roce 2010. Tam jej využívali členové záchranných sborů pro mapování škod a koordinaci svých operací. Z toho důvodu je navržena pro komunikaci s velmi špatným, nebo chybějícím připojením k internetu. To se projevuje zejména v komunikačním protokolu, který byl navržen pro účely omezení množství a komprese přenášených dat.

Základními funkcemi systému GINA je sledování vlastní polohy a poloh ostatních uživatelů, zakreslování obrázků do mapy pomocí stylusu, kdy jsou tyto obrázky převáděny na vektor a komprimovány, vytváření fotografií a jejich distribuce a komunikace pomocí výměny textových zpráv. Veškerá data jsou přitom zaznamenávána na serveru. Zaměření systému je od softwaru, který představuje tato práce, poněkud odlišné. Základním rozdílem jsou role uživatelů, kdy GINA umožňuje každému uživateli možnost vytvářet objekty na mapě, bez možnosti tuto tvorbu nějak centrálně řídit nebo ovlivňovat. Aplikace tedy slouží pro výměnu informací o stavu každého uživatele a o situaci v jeho okolí. Navrhovaný systém naproti tomu dovoluje vytvářet mapová data pouze jednomu typu uživatele, přičemž tento uživatel má roli určitého správce, který vytvářením těchto dat určuje, co má který uživatel v terénu dělat.

Jako nevýhoda aplikace GINA se zdá být volba implementačních technologií. Ta byla pravděpodobně ovlivněna cílením aplikace na soutěž Imagine Cup, kvůli které vývojáři zvolili pro implementaci výhradně technologie od společnosti Microsoft. To se vzhledem k zaměření softwaru na humanitární a neziskové organizace jeví jako problematické, protože například cena jedné licence pro využívaný databázový systém Microsoft SQL Server se stále pohybuje v desítkách tisíc korun. Z toho důvodu prezentovaný systém využívá pouze takových technologií, které jsou dostupné pod veřejnou licenci a je distribuován jako open-source.

2.3 Ostatní

Existuje množství dalších aplikací, které s prezentovaným systémem souvisí na základě nějaké vnitřní podobnosti. Společnou vlastností může být například využívání mapy nebo zaznamenávání geografické pozice. Pro příklad byly vybrány dvě aplikace. První je systém pro sledování pozice letadel, který má název Planefinder², a který lze vidět na obrázku 2.2 vlevo. Druhým je nástroj pro geocaching jménem Advanced Geocaching Tool³. Ten je na obrázku 2.2 vpravo.



Obrázek 2.2: Planefinder a AGTL

Advanced Geocaching Tool, neboli zkráceně AGTL, je aplikace pro mobilní telefony s operačním systémem Maemo a MeeGo. Jeho účelem je získávání dat, jako jsou popis,

²<http://planefinder.net/>

³<http://www.danielfett.de/internet-und-opensource,software,agtl-on-maemo-n900>

úroveň složitosti a obrázky, které usnadňují jeho uživatelům nalezení cache. Data jsou zobrazována na mapě, která je součástí aplikace a její podklady pocházejí z OpenStreetMap. Díky využití GPS jednotky mobilního zařízení je navíc možné vidět směr, kterým se má uživatel vydat a jaká je jeho vzdálenost.

Planefinder umožňuje sledovat pohyb letadel díky jejich zakreslování do mapy. Letadla jsou sledována pomocí technologie ADS-B. Ta nahrazuje starší způsob sledování letadel pomocí radaru. Využívá k tomu vestavěného zařízení na palubě letadla, které zaznamenává svoji polohu a spolu s doplňujícími informacemi o letadle ji vysílá do okolí. Aplikace Planefinder tato data sbírá a zobrazuje je na mapě. Ta využívá mapových podkladů Google Maps ve čtyřech variantách.

Kapitola 3

Mobilní platformy

Tato kapitola se v první části zabývá vývojem aplikací pro platformu Maemo [29]. Nejprve je popsána samotná platforma a je vysvětlen její vztah k novější platformě MeeGo [32]. Poté je zmíněn nástroj ScratchBox [31, 30], který slouží k překladu programů pro tyto platformy.

Dále kapitola pojednává o specifických rysech vývoje pro platformu Symbian [41]. Vysvětlen je jednak nástroj Smart Installer [34], sloužící k aktualizaci knihoven na cílovém zařízení, a jednak je popsán princip certifikátů [24, 40, 28].

Na závěr jsou zmíněny prvky knihovny Qt Mobility, které lze v projektu využít. Jedná se o součást této knihovny s názvem Location API, která poskytuje přístup ke geolokačním datům, značkám a QML¹ elementu pro zobrazení mapy [38].

Tato kapitola dále vychází z [17, 13, 1].

3.1 Maemo a MeeGo

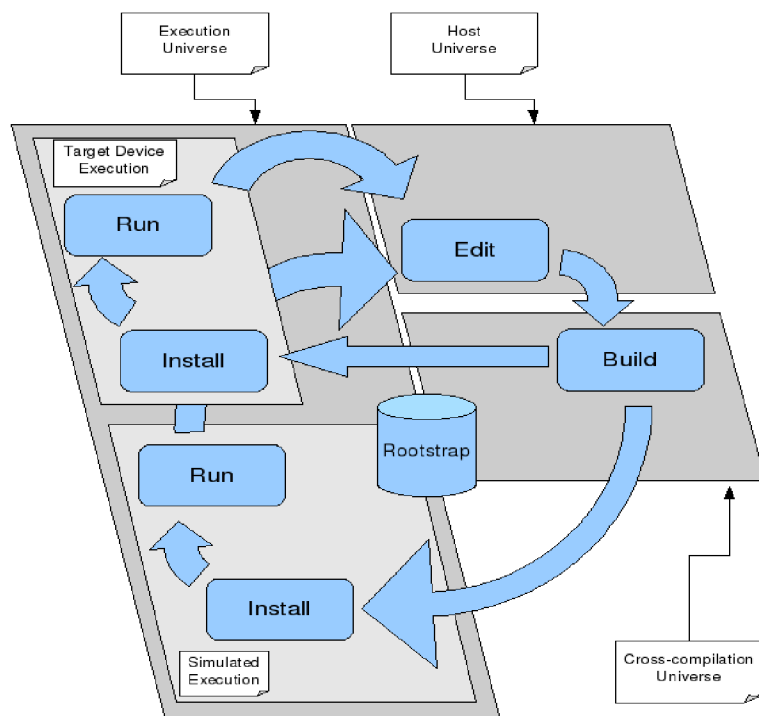
Maemo je operační systém určený pro mobilní zařízení postavený na debianu. Jedná se o otevřenou platformu, v současné době vyvíjenou z velké části komunitou open source vývojářů. Jádro systému je postaveno na Linuxu. O grafické prostředí se stará X Window využívající Xomap server a správce oken Matchbox. Grafické uživatelské rozhraní je vytvořeno pomocí knihovny Hildon, což je nadstavba nad grafickou knihovnou GNOME/GTK [29].

Současná verze se jmenuje Maemo 5 Fremantle. Tato verze se standardně vyskytuje na mobilním telefonu Nokia N900. Nová verze tohoto operačního systému byla zrušena a místo toho bylo rozhodnuto, že bude Maemo sloučeno s projektem společnosti Intel, zvaným Moblin, čímž vzniklo MeeGo. Základními částmi tohoto OS jsou MeeGo Core, MeeGo Touch Framework a Reference Handheld UX². Tento operační systém se v literatuře nejčastěji označuje jako MeeGo Handset, nebo MeeGo Handheld. Kromě této verze MeeGo existuje další, která se původně jmenovala Maemo 6, ale byla přejmenována na MeeGo Harmattan. Její základní odlišností je, že není založená na MeeGo Core, nicméně je s ním kompatibilní [32]. Tato verze pohání mobilní telefony Nokia N9 a Nokia N950 a je předmětem této práce.

Jak Maemo 5, tak MeeGo Harmattan mají velmi mnoho společného. Nejznatelněji je to vidět na vývoji aplikací, který je na obou platformách prakticky shodný. Důvodem je, že u obou jsou základními vývojovými nástroji Scratchbox a MADDE. Scratchbox je sada nástrojů, která slouží k usnadnění překladu programu pro vestavěná zařízení. Využívá tech-

¹Qt Modeling Language je na Javascriptu založený jazyk pro tvorbu uživatelských rozhraní v prostředí Qt.

²User eXperience



Obrázek 3.1: Schéma vývoje softwaru s nástrojem ScratchBox [31]

niku, která se nazývá cross-kompilace, což je označení pro překlad pro jiné architektury. To je přesně případ překladu pro mobilní zařízení, pro která je mnohem výhodnější kompilovat program lokálně s využitím cross-kompilace, a ten následně přeložený přemístit na cílovou architekturu a tam spouštět. Tento princip je graficky znázorněn na obrázku 3.1.

Vývojová architektura, na obrázku označovaná jako Host Universe, slouží k tvorbě programu. Překlad tohoto programu následně probíhá na Cross-compilation Universe, který obsahuje knihovny a hlavičkové soubory, označované jako Rootstrap, a nástroje potřebné pro kompilaci programů na cílové architektuře. Testování lze provádět buď přímo na cílovém zařízení, nebo jej simulovat na vývojové architektuře [31].

Jméno Scratchbox je odvozeno od anglického "Linux from scratch" a "sandbox". Důvodem k tomu je fakt, že programy v něm spouštěné běží v modifikovaném souborovém systému vytvořeném pomocí unixového systémového volání chroot. To slouží ke změně kořenového adresáře pro daný proces, díky němuž je tento izolován od svého okolí a nemůže způsobit velké škody. Scratchbox je tedy softwarový balík, který umožňuje izolovat procesy od jejich okolí. Dále poskytuje prostředky pro cross-kompilování a dovoluje spouštět zkompileované programy na mobilních zařízeních, nebo lokálně na emulátoru [30]. Jako emulátor pro zařízení s operačním systémem Maemo slouží aplikace QEMU, což je otevřený emulátor a virtualizátor, který umožňuje spouštět programy a operační systémy určené pro jiné platformy, ne na kterých sám běží. QEMU umožňuje poměrně přesně simulovat chování skutečného zařízení.

Tato přesnost je však vykoupena značnou pomalostí celé aplikace a dlouhou dobou trvání jejího startu. Z tohoto důvodu se v nových verzích Qt SDK využívá více aplikace Simulator, která je významně rychlejší. Důvodem je, že překládá a spouští aplikace jako desktopové, pouze s tím rozdílem, že využívá knihovny, které jsou dostupné na cílovém

zařízení. Výstup následně zobrazuje v rozlišení, jakým toto zařízení reálně disponuje.

Pro vývoj na Maemo respektive MeeGo byl vydán nástroj Maemo Application Development and Debugging Environment, neboli MADDE. To je nástroj, který vychází z výše jmenovaného Scratchboxu. Jeho účelem je usnadnit cross-kompilaci do takové míry, že programátor nemusí vůbec vědět, že probíhá. MADDE se využívá pro překlad programů z prostředí Qt SDK. Aby bylo možné jej využívat, je nutné na zařízení nainstalovat aplikaci mad-developer. Díky ní je pak možné zasílat, instalovat, spouštět a ladit v SDK přeložené aplikace [29].

3.2 Symbian

Symbian je třetím mobilním operačním systémem, který byl vybrán jako cílová platforma. Důvodem je, že pro překlad pro něj je nativně podporován v Qt SDK a tudíž jsou změny, které je nutné při portování ve zdrojovém kódu provést, minimální. Princip kompilace je prakticky shodný jako v případě dvou předchozích linuxových platform [41]. Specifičnost Symbianu tkví jednak ve způsobu aktualizace knihoven na mobilním zařízení a jednak v certifikační politice této platformy.

Pro zajištění toho, aby překládaná aplikace vždy fungovala na cílovém zařízení, je možné u platformy Symbian zvolit jednu ze tří metod. První je stažení a instalace balíků s aktuálními knihovnami z internetu ve formátu SIS³, což je archiv obsahující binární soubory aplikace. Druhým je spuštění aplikace Aktualizace Softwaru přímo v telefonu a třetí je překlad aplikace s příložením Smart Installeru.

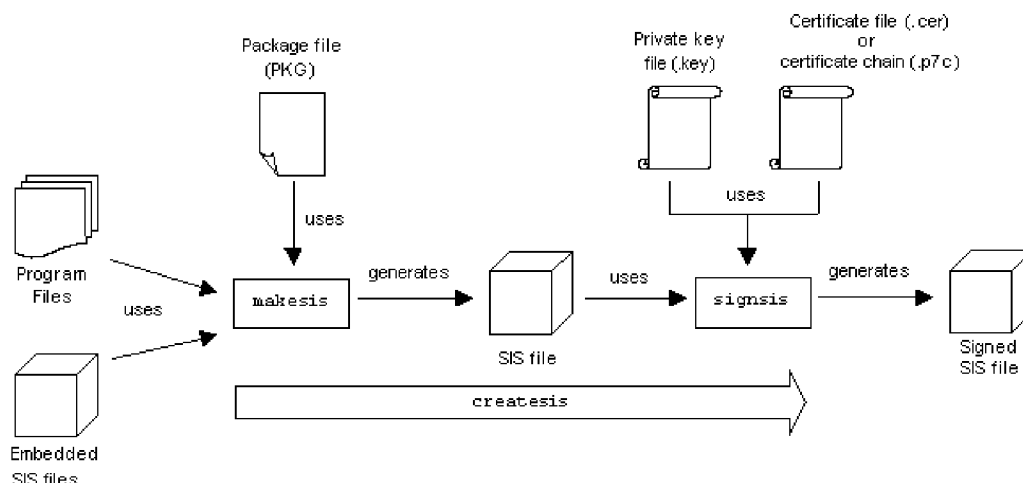
Smart Installer je řešením pro aplikace třetí strany, aby se tyto vždy bez problému nainstalovaly a spustily na cílovém zařízení a měly k dispozici všechny knihovny ve verzi, pro kterou byly napsány, aniž by tyto knihovny musely být vždy přidány k instalačnímu balíku aplikace. Tento nástroj funguje tak, že jsou v okamžiku kompilace zjištěny všechny závislosti aplikace z konfiguračního souboru projektu. Ty jsou uloženy ve Smart Installeru, který je následně pomocí techniky bootstrap přiložen ke kompilované aplikaci. Ta může být poté nainstalována na jakémkoli zařízení. Během instalace Smart Installer spustí za pomoci aplikace Smart Installer ADM⁴ kontrolu, zda jsou na zařízení přítomny všechny knihovny v požadovaných verzích a pokud ne, tak v rámci instalace aplikace zahájí jejich stahování. Jakmile jsou aktualizovány všechny knihovny, spustí se instalace samotné aplikace [34]. Za účelem ladění existuje pro platformu Symbian aplikace CODA Debugging Agent, která nahradila starší Symbian TRK. Jedná se o obdobu výše zmíněné aplikace mad-developer, která oproti svému předchůdci navíc umožňuje ladění na zařízeních, která jsou připojena přes wi-fi bez nutnosti jejich připojení přes USB kabel.

Druhým specifikem platformy Symbian jsou certifikace. Kvůli architektuře Symbian Platform Security jsou v rámci této platformy omezeny citlivé funkce, kterým se říká capabilities. Aby k nim získal vývojář přístup, musí mít k dispozici takový certifikát, který je prověřen pro užívání odpovídající funkce. Základní funkcionalita, dostupná na platformě, vyžaduje pouze certifikát self-signed, o který nemusí vývojář žádat, ale pokud chce například číst kontakty nebo zprávy uživatele, potřebuje už ReadUserData Capability a k ní odpovídající prověření svého certifikátu. Celkově platforma Symbian obsahuje 20 různých omezení a jejich výčet spolu s popisem lze nalézt na [24].

Pro tvorbu pokročilejších aplikací je tedy nutné žádat o vývojářské certifikáty. Ty jsou

³Software Installation Script

⁴Application Dependency Manager



Obrázek 3.2: Schéma certifikace aplikace při překladu u OS Symbian [28]

v současné době přidělovány na čtvrt roku a aby se k nim vývojář dostal, musí buď využít bezplatné služby na [40], nebo se musí zaregistrovat jako vývojář a následně žádat o aktualizovaný certifikát prostřednictvím emailu.

Obrázek 3.2 ukazuje proces certifikace při tvorbě instalačního balíku aplikace pro platformu Symbian. Nástroj `makesis` vytváří necertifikovaný soubor SIS s aplikací. Ten je převzat nástrojem `signsis`, který pomocí dvojice souborů `.key` a `.cer` balík certifikuje a vytvoří tak podepsaný instalátor aplikace [28].

3.3 Qt Mobility

Qt Mobility je knihovna pro zařízení běžící na platformách Maemo, MeeGo a Symbian, která poskytuje nástroje pro práci s pokročilými mobilními technologiemi, které jsou na těchto zařízeních dostupné. V rámci této práce je využívána součást Qt Mobility zvaná Location API. Ta se skládá ze tří částí, a sice: určování pozice, značek a map s navigací. Určování pozice poskytuje ze satelitů získané údaje o zeměpisné šířce a délce, nadmořské výšce, azimutu a rychlosti. Značky umožňují získávat a ukládat důležitá místa a mapy a navigace dovolují využívat mapových podkladů a vypočítávat trasy cest [38].

Geolokační data získaná pomocí Location API dovolují přesně specifikovat pozici zařízení na povrchu Země pomocí zeměpisné šířky a délky. Tato data lze získat pomocí několika metod, z nichž nejznámější je využití GPS⁵. To je veřejně dostupný systém, který za použití radiových vln přijatých od satelitů na orbitě počítá přesné souřadnice, na nichž se přijímač nachází. Druhou technikou určování geografické pozice je využití Cell ID, což je identifikátor BTS⁶, což je základnová stanice v síti GSM, k níž se připojují mobilní telefony, ke které je mobilní zařízení momentálně připojeno. Pomocí této znalosti a znalosti o tom, kde se daná BTS nachází, lze odhadnout, kde se nachází k ní připojené zařízení [38]. Další možností je využití techniky A-GPS⁷, která slouží ke zpřesnění a zrychlení vyhledání vlastní pozice.

⁵Global Positioning System

⁶Base Transceiver Station

⁷Assisted GPS

Funguje tak, že si zařízení stáhne z internetového asistenčního centra přesné záznamy o dráhách a palubních časech všech GPS družic a zaměří se na signál konkrétní družice, která se nachází nad ním. V rámci Location API lze programově zvolit jak interval, po kterém se pozice automaticky aktualizuje, tak i metodu, jakou jsou data získávána.

Značky v Location API umožňují vytvářet, získávat, obnovovat a mazat významná místa z volitelných datových úložišť. Značka zde představuje určité místo nesoucí nějaký význam, jako může být pozice uživatele. Každá značka může mít několik atributů, například jméno, doplňující text, pozice, ikona a radius. Pomocí těchto atributů lze následně značky filtrovat a vyhledávat [38].

Kapitola 4

Technologie

Tato kapitola nejprve představuje mapové služby [9, 35], které lze v současné době volně využít, srovnává je a vybírá tu, která se v projektu hodí nejvíce. V další části diskutuje vhodnou volbu reprezentace mapových objektů, které se na této mapě budou zobrazovat [14, 39]. Poslední část se pak zabývá volbou vhodného Systému řízení báze dat pro tato data [37, 36].

4.1 Mapové podklady

Mapové podklady je možné reprezentovat jak rastrově, tak i vektorově. Vektorová data mají tu výhodu, že zabírají menší množství paměti a je tak výhodnější je posílat přes internet. Nevýhodou může být, že je data při přijetí nejprve nutné renderovat, což má nároky na výkon. Další nevýhodou je nemožnost využití satelitních snímků pro zobrazení mapy. Poskytovatelé mapových služeb v dnešní době nabízejí prakticky výhradně mapové podklady v rastrové formě. Navíc prakticky neexistují volně dostupné implementace renderovacího nástroje pro vykreslení map ve vektorovém formátu. To je důvodem, proč se dále v této podkapitole bude uvažovat pouze o poskytovatelích rastrových mapových podkladů.

Poskytovatelů mapových podkladů je v dnešní době poměrně velké množství. Příkladem mohou být Google Maps, OpenStreetMap [35], Bing Maps a Nokia Maps. Všechny mají z pohledu této práce velmi podobné vlastnosti. Proto jsou v této podkapitole na principu OpenStreetMap vysvětleny některé jejich detaily. Ve zbytku podkapitoly jsou pak představeny ostatní alternativy a z nich je vybrána ta nejvhodnější. Při psaní této podkapitoly bylo čerpáno z [9].

OpenStreetMap je projekt, který se věnuje tvorbě volně dostupných map celého světa. Tvorba těchto map se provádí buď za pomoci gps zařízení při pohybu v terénu, využíváním leteckých snímků, ruční tvorbou z rastrových podkladů, nebo jsou využívány již existující volně dostupné mapy. Na rozvoji projektu se podílejí i vládní organizace. Nejlépe zmapovanými částmi světa jsou Evropa a USA.

OpenStreetMap jsou tvořeny několika druhy geografických primitiv. Nejjednodušším je bod, dalším je cesta, která je složena z několika bodů, a třetím je plocha tvořená třemi a více body, které vytyčují uzavřenou oblast. Posledním typem je plocha s dírou uvnitř.

Objekty, jež tvoří nejčastěji sami uživatelé, se ukládají do databáze jako geografická vektorová data. Pro zobrazování se tato data následně převádí do grafické podoby. Vznikají tak čtvercové rastrové dlaždice o straně 256 pixelů. Nejpoužívanějšími renderovacími ná-

stroji jsou Osmarender¹ a Mapnik². Takto vytvořené rastrové mapy jsou volně dostupné na webovém serveru, kde jsou uloženy podle své horizontální a vertikální souřadnice a úrovně přiblížení [35].

Z dalších volně dostupných mapových služeb lze jmenovat Google Maps, Bing Maps a Nokia Maps. Ty jsou často kromě motivace svých tvůrců postaveny na podobných principech jako OpenStreetMap, a proto už nebudou detailně popisovány. Výhodou map od společnosti Google jsou na některých místech větší detaily než u OpenStreetMap. Na druhou stranu fakt, že podklady pro OpenStreetMap vytváří komunita uživatelů, občas umožňuje, že jsou tyto mapy aktuálnější, než mapy konkurence. Tak tomu bylo například při zemětřesení na Haity. Další výhodou je, že nabízejí tři druhy pohledu, kromě normálního ještě satelitní a terénní. Nevýhodou je nutnost registrace u Google a některá licenční omezení. Bing mapy od společnosti Microsoft jsou svým zaměřením prakticky stejné jako mapy Google. Mírnou nevýhodou může být to, že jsou ze všech jmenovaných nejnovější a tudíž nejméně otestované. Totéž se může týkat i map Nokia, které jsou sice k dispozici déle, ale nemají takový počet uživatelů, jako ty od Google a nejsou proto tak pečlivě prověřeny. Tato námitka se nicméně na území České republiky nijak výrazně nepromítá.

Všechny zmíněné mapové služby jsou si víceméně podobné. Jejich výběr tedy do značné míry záleží na osobní preferenci, nebo na poskytovaném API a dodatečných službách, které podporují. Jako nejjednodušší volba pro implementaci se jeví mapy Nokia, protože mají přímou podporu ve vývojářských nástrojích této společnosti, kde je nabízen hotový mapový widget pro jejich zobrazování. Pro lepší zhodnocení je ale nutné otestovat více řešení v praxi.

4.2 Zobrazení mapy

Navrhovaný software vyžaduje pro optimální funkčnost vhodný nástroj pro zobrazování mapových podkladů. Pro posouzení této problematiky byl implementován vlastní mapový widget. Ten byl následně porovnán s dalšími třemi widgety za účelem výběru nejvhodnějšího prostředku pro zobrazování mapy v aplikaci. Tato podkapitola se zabývá jejich analýzou a popisem experimentování s nimi.

Podmínkou při výběru mapové komponenty vhodných pro testování byla kompatibilita s využitým frameworkem Qt. Z toho důvodu byla zvolena komponenta využívaná v aplikaci Advanced Geocaching Tool³, která byla popsána v předcházející kapitole. Jako druhý byl zvolen QMapControl, což je volně dostupný otevřený widget, který je součástí softwaru Merkaator⁴. To je mapový editor využívaný v rámci projektu OpenStreetMap. Třetím mapovou komponentou je standardní mapový widget knihovny Qt Mobility, který se nazývá QML Map⁵.

Kritéria pro hodnocení mapových komponent jsou dvě. První je množství podporovaných nástrojů využitelných v rámci této práce. Jedná se zejména o API pro vykreslování grafických objektů do mapy a podporované typy mapových podkladů. Druhým kritériem je rychlost vykreslování mapových podkladů.

Komponenta vytvořená v rámci experimentů s mapovými technologiemi QtMapWidget využívá služeb OpenStreetMap. Důvodem je dostupné a dobře zdokumentované API a minimální omezení z hlediska využívání dostupných dat. Cílem při implementaci widgetu byla

¹<http://wiki.openstreetmap.org/wiki/Osmarender>

²<http://mapnik.org/>

³<http://www.danielfett.de/internet-und-opensource,software,agtl>

⁴<http://merkaator.be/>

⁵<http://doc.qt.nokia.com/qtmobility/qml-map.html>

zejména rychlost vykreslování, již je dosaženo pomocí ukládání mapových dlaždic do paměti. Díky tomu je možné využívat jednou stažených dlaždic opakovaně, bez nutnosti je znovu stahovat. Ukládání dlaždic je sice pomalejší, avšak v konečném důsledku je používání mapy velmi plynulé. V rámci implementace byla též zahrnuta funkce pro vykreslování bodů do mapy spolu s doplňujícím textem. Zvažována byla i možnost vytvořit funkce pro vykreslování čar a polygonů. Ty se však ukázaly být implementačně poměrně náročné a proto bylo od nich upuštěno. Velkou výhodou vlastního řešení je možnost úplného přizpůsobení widgetu potřebám aplikace. Je tak možné například stáhnout si potřebná data z menšího území a ta pak využívat bez nutnosti stahovat je až v době potřeby. Nevýhodou je nutnost implementovat komponentu od začátku, což v kontextu ostatních řešení nemusí být výhodné.

Mapový widget, který využívá AGTL, je vytvořen v technologii QML, díky čemuž je velmi snadno rozšiřitelný. Jako mapové podklady využívá opět OpenStreetMap a podle autora umožňuje zobrazovat po rozšíření i mapové podklady společnosti Google. Výhodou je podpora rotace mapy při změně orientace uživatele s telefonem. Další vlastností vhodnou pro využití v této práci je vykreslování polohy uživatele na mapě. Jako nedostatečné se jeví dostupné rozhraní pro vykreslování grafických objektů, které podporuje pouze jednoduché body. Widget je možné rozšířit o offline mód.

Třetí komponentou pro zobrazování mapy, s níž byly prováděny experimenty, je QMapControl. Práce na tomto projektu skončily již v roce 2009, kvůli čemuž je dnes už poměrně zastaralý a tak z původně podporovaných mapových podkladů jsou již funkční pouze OpenStreetMap a Yahoo Maps. Díky jeho otevřené distribuci však byla zprovozněna i podpora Google Maps. Velkou výhodou je podpora víceúrovňového vykreslování dlaždic, která umožňuje prokládat více mapových podkladů přes sebe. Prvek má k dispozici jednoduché API, pomocí kterého lze do mapy zakreslovat body, čáry a obrázky. S využitím zdrojových kódů zmíněného projektu Merkaator lze však možnosti widgetu dále rozšířit. Zastaralost projektu se projevuje také v tom, že je celý kompletně vytvořen v C++, což znesnadňuje jeho propojení s novější technologií QML. Offline mód není podporován.

Posledním mapovou komponentou je QML Map, která využívá podkladů Nokia Maps. Díky tomu, že je tento widget přímo součástí knihovny Qt Mobility, je jeho implementace v projektu velmi snadná. Ze všech testovaných widgetů nabízí nejbohatší API pro vykreslování grafických objektů do mapy. Vykreslit je možné obdélníky, kružnice, body, čáry, polygony, obrázky i text. Výhodou je propojení s jinou částí Qt Mobility, která se jmenuje Landmarks. Toto spojení umožňuje v C++ vytvářet značky, které jsou okamžitě zobrazovány na mapě. Ačkoli je podporováno pouze využití podkladů společnosti Nokia, umožňuje zobrazit 3 typy dlaždic. Kromě klasických jsou to i terénní a satelitní. Podle specifikace má být možné podporovat zobrazování map v offline režimu. Tuto vlastnost se však podařilo zprovoznit pouze ve starší verzi knihovny Qt Mobility a v nové verzi nefunguje. Je však pravděpodobné, že bude tato chyba s další verzí knihovny opravena.

Ze zmíněných pozitiv a negativ testovaných mapových komponent a z rychlostí jejich odezvy plyne, že i přes aktuálně nefunkční offline režim se jako nejvhodnější alternativa pro implementaci v rámci představovaného systému jeví standardní komponenta QML Map.

4.3 Grafická reprezentace dat

Systém slouží k orientaci lidí v prostoru. K tomu se využívá mapa, do níž jsou zakreslovány grafické objekty. V této podkapitole jsou postupně zváženy některé alternativy, jak tyto objekty vhodně reprezentovat.

Mapová data je možné popsat rastrově, nebo vektorově. Výběr způsobu reprezentace je jedním ze základních rozhodnutí, které je nutné při návrhu softwaru využívajícího mapy učinit. V případě navrhovaného systému je cílem zakreslovat jednotlivé geometrické objekty do mapy, tvořené rastrovými dlaždicemi, jak bylo zmíněno na začátku této kapitoly. Reprezentace těchto objektů jako rastrů může vyžadovat jejich posílání přes síť a ukládání jako multimediálních souborů. Takto realizovaný objekt má přesně dané souřadnice, kam je nutné jej vykreslit a dále s jeho geometrií není možné manipulovat.

Jako logická se proto jeví vektorová reprezentace dat, která nachází opodstatnění ve všech částech systému. Při vytváření mapových objektů je účelné tyto tvořit jako množiny bodů, s nimiž lze i po vytvoření dále manipulovat. Reprezentace těchto bodů je možná v textové podobě, zabírá menší množství paměti a lze ji reprezentovat v databázových systémech, které mají podporu pro geometrická data. V těchto systémech lze navíc nad těmito daty tvořit dotazy.

Jako formát vektorových dat je možné zvolit ten, který umožňuje vygenerovat použitý systém řízení báze dat. Například systém PostGIS dovoluje generovat geografická data ve formátu Shapefile [5], což je formát pro reprezentaci bodů, čar a oblastí. Tvořen je třemi různými soubory. Tyto soubory nejsou čitelné člověkem. Využívají se pro výměnu dat mezi databázemi, nikoli pro přímé zobrazení na mapě. Proto je volba této reprezentace pro navrhovaný systém nevhodná.

Další možností je volba reprezentace v textové podobě například s využitím značkovacích jazyků JSON nebo XML. Tato reprezentace je snadno převoditelná na jakýkoli požadovaný formát. Jako vhodné varianty jsou v následujících odstavcích popsány GML a SVG. Zvážena je také volba vlastního formátu dat.

GML neboli Geography Markup Language je na XML postavený formát pro reprezentaci geografických prvků. Je využíván jako otevřený formát pro výměnu geografických dat a jako modelovací jazyk v geografických informačních systémech. Dokument v GML je složen z aplikačního schématu, který popisuje dokument a samotných geografických dat [14]. Pro tuto práci se jeví jako vhodný z hlediska reprezentace geometrie mapových dat, ale postrádá jakýkoli popis jejich vzhledu. Spolu s GML dokumentem by tedy bylo nutné přenášet ještě metainformace obsahující například barvu objektu.

SVG⁶ je formát založený na XML. Umožňuje popisovat vektorové entity a zároveň jim přiřazovat styl s využitím CSS, jako je barva nebo textura [39]. SVG podporuje 3 typy objektů. Jednak jsou to vektorové tvary, dále rastrové obrazy a textové objekty. Všechny tyto objekty lze seskupovat. Tyto vlastnosti spolu s existencí specializované knihovny v prostředí Qt dělají z SVG ideální formát pro reprezentaci prostorových dat.

Existuje ještě řada dalších způsobů, jak reprezentovat vektorová data. Většina jich ale nepřináší v kontextu této práce nic, co by některý ze zmíněných formátů nenabízela. Jinou možností by bylo navrhnout a implementovat vlastní řešení. Toto řešení by ovšem muselo řešit podobné problémy, jaké řeší už formát SVG, takže by v konečném důsledku bylo spíše kontraproduktivní.

4.4 Databázové systémy

V předchozí podkapitole byl zvolen způsob, jakým budou reprezentována mapová data v grafické podobě. Tato podkapitola se zabývá jejich uchováním. Jedním z kritérií pro výběr vhodného systému řízení báze dat je to že, musí být distribuován pod veřejnou licenci.

⁶Scalable Vector Graphics

Z tohoto důvodu byly vybrány systémy MySQL [33], SQLite [25], PostgreSQL [37] a jeho rozšíření PostGIS [36]. Systémy jako MS SQL, nebo Oracle nejsou ve výčtu uvažovány právě z důvodu jejich komerčního zaměření. A to i přesto, že je druhý jmenovaný díky široké podpoře geografických datových typů a funkcí nad nimi pro navrhovaný systém velmi vhodný.

MySQL je otevřený relační databázový systém šířený jak pod bezplatnou licencí GPL, tak i pod licencí komerční. Je snadno implementovatelný a je možné jej využít jak na systémech s operačním systémem Windows, tak i na systémech s Linuxem. Systém je plně vícevláknový a dokáže tak využívat více procesorů, či jader. Ke každému úspěšně vytvořenému připojení je přiděleno vlákno. Všechny dotazy jsou pak vykonávány uvnitř tohoto vlákna. Díky tomu, že si systém jednotlivá vlákna uchovává ve vlastní cache, není nutné tato ukončovat a znovu vytvářet s každým novým připojením [33].

Ačkoli nové verze systému podporují triggery, pohledy i uložené procedury, nepřináší z pohledu této práce kromě rychlosti a jednoduchosti žádné zvláštní výhody. Pro ukládání prostorových dat by bylo nutné zvolit vlastní reprezentaci a nebylo by možné nad těmito daty vytvářet dotazy.

SQLite je databáze, která je celá tvořena jedním souborem na disku. Nevyžaduje tedy žádný databázový server, na němž by musela běžet. Databázový soubor může být sdílen mezi různými platformami. Výhodou této databáze je, že má velmi malé nároky na hardware zařízení, na němž běží. Jak je uvedeno na stránkách autorů [25], zabírá standardní knihovna pro jazyk C 25kb paměti. Díky tomu se tato databáze hodí pro nasazení na mobilních zařízeních. Platformy, pro něž je prezentovaný systém vytvářen, navíc nabízejí pro SQLite nativní podporu. Jak je popsáno v sekci 5.2, nevyžaduje stávající řešení využití databáze na mobilním klientu. V případě implementace rozšíření, které jej vyžaduje, je SQLite však vhodným kandidátem.

Mezi zásadní nevýhody SQLite lze zařadit fakt, že systém nepodporuje kontrolu typů, takže umožňuje například zadat řetězec znaků do sloupce, kde je očekáváno číslo. Jedinou výjimku tvoří sloupec pro primární klíč, který očekává unikátní celé číslo. Další nevýhodou je chybějící kontrola platnosti cizích klíčů v tabulkách. SQLite nemá vlastní reprezentaci pro typ data a času, a tak je nutné konvertovat tyto údaje na číslo v režii programátora. Z hlediska podpory geografických typů tato databáze nenabízí žádné výhody a pro nasazení na straně koordinátora se nejvíce jeví jako vhodná.

PostgreSQL je objektové orientovaný relační databázový systém. Je vydáván pod licencí MIT a tudíž je jak otevřený, tak i volně dostupný bez licenčních poplatků. Primárně je vyvíjen pro Linux, ale existují i verze pro Windows. PostgreSQL má zabudovanou podporu B+tree, hash, GiST a GiN indexů. Navíc dovoluje vytvářet uživatelům jejich vlastní indexy. Dále podporuje jak triggery, tak i pohledy a uložené procedury. Nabízí dokonce podporu datových typů pro reprezentaci geometrických objektů, jako je bod, čára i polygon, nepřináší však žádné výhody pro práci s geografickými objekty [37].

PostgreSQL je tedy uspokojivým řešením pro reprezentaci dat v systému. Reprezentace prostorových dat pouze jako geometrických, nikoli však geografických, je vážným omezením, které nedovoluje tento systém využít v jeho základní verzi.

PostGIS je rozšířením databázového systému PostgreSQL o komplexní reprezentaci geografických objektů. Základní typy zahrnují body, čáry i polygony a jejich kolekce. PostGIS umožňuje tvorbu dotazů na průnik objektů, výpočet vzdálenosti i výměr překryvu oblastí. Jako prostorový index využívá R-tree [36]. Reprezentace geografických dat je tedy v systému realizována na úrovni, která plně vyhovuje požadavkům této práce. Zmíněné vlastnosti PostGIS činí vhodným kandidátem na využití v systému.

4.5 Komunikace

Tato podkapitola ve své první části provádí srovnání protokolů, které mohou sloužit pro výměnu dat mezi mobilními klienty a serverem. Mezi protokoly, které jsou vzhledem k této práci relevantní vybírá HTTP [6, 7], REST [16], SOAP [21, 22, 23] a XMPP [18]. Základními kritérii pro srovnání protokolů jsou bezpečnost a vhodnost pro účely aplikace, kterými jsou zejména způsob autentizace uživatelů a podpora pro přenos geografických dat. Na základě těchto kritérií následně vybírá protokol XMPP pro implementaci požadované funkcionality. Druhá část podkapitoly se tedy věnuje konkrétnímu popisu tohoto protokolu. Podkapitola dále čerpá z [2, 11].

HTTP, neboli HyperText Transfer Protocol, je textový protokol pro přenos dat na internetu. Jeho hlavním účelem je přenos souborů HTML nebo XML, ale pomocí rozšíření MIME⁷ je možné jím posílat i jiná, například multimediální, data. Soubory jsou identifikovány pomocí URL⁸, což je adresa, která jednoznačně identifikuje daný soubor na internetu.

Protokol funguje na principu Klient/Server, kdy klient většinou pošle na server požadavek, na základě kterého je na serveru vytvořena odpověď a ta je poslána zpět klientovi. Jedná se o bezstavový protokol, takže si server neukládá žádné informace o komunikaci, jako je například přihlášení určitého klienta k aplikaci. Server nedokáže rozeznat, zda spolu dva předchozí požadavky nějak souvisí. Kvůli tomu byl protokol rozšířen o HTTP Cookies [8], které umožňují uchovávat si informace o stavu komunikace u klienta.

Z hlediska bezpečnosti je nutné využívat rozšíření protokolu, s názvem HTTPS⁹. To umožňuje zabezpečit spojení mezi klientem a serverem před odposloucháváním a podvržením dat. Navíc dokáže ověřit identitu protistrany. HTTPS šifruje data pomocí SSL¹⁰ a TLS¹¹ [7]. Pro zajištění bezpečnosti musí požadavky mít zadánu konkrétní metodu. Ta určuje, jak mají být na serveru zpracovány a jaká odpověď má být poslána zpět. Některé metody, jako GET nebo OPTIONS jsou označovány jako bezpečné. Neměly by tedy způsobit změnu stavu dat na serveru. V praxi však není jejich zpracování na serveru ničím omezeno, takže mohou vyvolat vedlejší účinky. Jiné metody, jako TRACE jsou považovány za nebezpečné, protože mohou útočníkovi poskytnout informace o serveru, nebo posloužit při obcházení bezpečnostní kontroly [6].

Autentizace uživatele pomocí protokolu HTTP je dvojího druhu. První, WWW - Authenticate [6], funguje tak, že když nezašle klient serveru požadavek s platným jménem a heslem, zašle mu server odpověď s hlavičkou 401 Unauthorized. Klient poté získá přihlašovací údaje od uživatele a zašle je na server. Tam jsou ověřeny buď v aplikaci, nebo pomocí funkce serveru. Tento typ autentizace má několik nevýhod. Kvůli bezstavovosti protokolu musí být heslo zasíláno s každým požadavkem a spojení tak musí být šifrováno. Značně problematický je mechanismus odhlášení. Z těchto důvodů je tento způsob v prezentované aplikaci nevhodný. Server si totiž musí uchovávat přesné informace o tom, kteří uživatelé jsou v dané chvíli připojeni.

Jako lepší způsob se jeví autentizace formulářová [10]. Ta funguje tak, že server spojí platné přihlášení s aktuální session. Klient pak zasílá pouze session token místo uživatelského jména a hesla. Odhlášení probíhá jednoduše tak, že server buď zruší propojení mezi přihlášením a session, nebo rovnou zruší celou session. Zaslání jména a hesla na začátku

⁷Multipurpose Internet Mail Extensions

⁸Uniform Resource Locator

⁹Hypertext Transfer Protocol Secure

¹⁰Secure Socket Layer

¹¹Transport Layer Security

spojení je opět nutné šifrovat. Tento způsob autentizace je pro aplikaci lepší a je možné jej využít. Jako nedostatek lze považovat nutnost řešit autentizaci uživatelů na straně implementovaného serveru, což zvyšuje režii na serveru.

Protokol HTTP nenabízí žádná hotová řešení pro distribuci geografických dat. Z toho vyplývá, že je v případě jeho nasazení nutné navrhnout a implementovat rozšíření tohoto protokolu, které by umožňovalo tato data reprezentovat a přenášet. Modelová situace publikace polohy klienta na server v navrhovaném systému pomocí HTTP metody GET může vypadat jako v příkladu 4.1.

```
1 GET /updateposition?jid=Novak123456&lat=49.817&lon=15.472 HTTP/1.1
```

Příklad 4.1: Publikování polohy pomocí HTTP

Jedním ze zmíněných nedostatků HTTP je, že metoda může vyvolat na straně serveru nežádoucí účinky. Tento problém řeší Representational State Transfer, neboli REST, což je architektura, jež definuje množinu principů, s užitím kterých lze vytvořit síťovou službu zaměřenou na data. REST není specifikován pro jediný protokol, nicméně se nejčastěji implementuje ve spojení s HTTP, což je kombinace, která se pro účely této práce hodí nejvíce [16].

Základním pojmem u REST je zdroj. Jako zdroj může být označena každá informace, kterou lze pojmenovat. Zdrojem mohou být obrázky, služby, nebo dokumenty, případně jejich kolekce. Každý zdroj je reprezentován jednoznačně pomocí URI¹². Server na klientovy požadavky vrací reprezentaci zdroje, což je v podstatě stav zdroje ve chvíli, kdy jej klient požaduje. Reprezentace je v tomto smyslu tedy snímek zdroje v konkrétním čase. V praxi to znamená například to, že jsou řádky tabulky v databázi reprezentovány jako elementy v XML¹³ dokumentu.

Z hlediska bezpečnosti architektura REST využívá vlastností protokolu, pro který je implementována. Pro protokol HTTP však přináší jednu podstatnou výhodu a tou je užívání jeho metod explicitně. To znamená, že použití každé metody serveru jasně říká, co má se zdrojem dělat. Nemůže se tedy stát, že by například metoda GET způsobila změnu stavu dat na serveru. Metody, které klient používá jsou POST pro vytváření dat na serveru, PUT pro jejich editaci, GET pro získání a DELETE pro smazání. Metody PUT, GET a DELETE jsou idempotentní, což znamená, že pokud by stejný požadavek s nimi dorazil na server vícekrát, tak způsobí změnu pouze jednou. Příklad 4.2 ukazuje, jak příklad, ve kterém klient posílá aktualizaci své pozice na server, vypadá s využitím architektury REST.

```
1 PUT /positions/Novak123456 HTTP/1.1
Host: myserver
3 Content-Type: application/xml
<?xml version="1.0"?>
5 <position>
  <lat>49.817</lat>
7  <lon>15.472</lon>
</position>
```

Příklad 4.2: Publikování polohy pomocí REST

Použití PUT v předchozím případě je explicitní v tom smyslu, že identifikuje zdroj, který má být editován pomocí URI požadavku a tak prakticky přenáší novou reprezentaci zdroje z klienta na server v těle požadavku PUT, místo aby přenášel atributy zdroje přímo

¹²Uniform Resource Identifier

¹³Extensible Markup Language

v URI [16]. To přináší oproti HTTP větší bezpečnost a zároveň umožňuje data lépe číst v nezpracované formě uživatelem. Na druhou stranu to stále nepřináší žádné již hotové řešení pro zasilání geografických dat a to je nutné si, stejně jako v předchozím případě, vytvořit vlastní.

REST je bezstavový, takže si neukládá informace o připojení a klient tak musí v každém dotazu zasílat své přihlašovací údaje, nebo využívat session. REST se často využívá ve spojení s otevřeným autentizačním protokolem OAuth [19]. V tomto ohledu pro něj tedy platí totéž, co pro samotný protokol HTTP.

Jinou možností je využití na XML založeného protokolu SOAP, což je zkratka pro Simple Object Access Protocol. Ten standardně podporuje několik typů volání funkcí služeb, z nichž nejznámější je Remote Procedure Call. Jako transportní vrstvu využívá opět protokol HTTP. Z toho vyplývají shodné vlastnosti, pokud jde o zabezpečení protokolu. Pro autentizaci se využívá transportní vrstvy, nebo ji lze provádět na úrovni SOAP. V prvním případě platí stejné vlastnosti a omezení jako u protokolu HTTP. V druhém případě se autentizační údaje zadávají jako parametry v hlavičce SOAP zprávy, přičemž jsou často následně nahrazeny vygenerovaným tokenem. Protokol je bezstavový a tudíž pro udržování kontextu uživatele vyžaduje vytváření session.

Nevýhodou oproti architektuře REST je z hlediska bezpečnosti fakt, že pro volání je využívána HTTP metoda POST a tudíž není bez nahlédnutí do těla zprávy například dopředu jasné, jestli žádost má pouze získat data ze serveru, nebo smazat celou tabulku. Nutnost prohledání těla zprávy má pak vliv na rychlost systému. Pokud jde o podporu geografických dat, platí pro SOAP totéž co pro REST, a sice že není pro tuto potřebu definován žádný standard. Příklad 4.3 ukazuje, jak může vypadat publikování polohy při použití protokolu SOAP.

```
POST /GeoLocation/GeoLocation.asmx HTTP/1.1
2 Host: www.coordinator.net
Content-Type: application/soap+xml; charset=utf-8
4 Content-Length: length

6 <SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
8 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
10   <m:InsertPosition xmlns:m="urn:x-example:services:Position">
     <lat>49.817</lat>
12     <lon>15.472</lon>
     </m:InsertPosition>
14   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Příklad 4.3: Publikování polohy pomocí SOAP

Z výše uvedeného vyplývá, že jak protokol HTTP, tak i protokoly z něj vycházející, mohou být pro navrhovaný systém využity. Všechny nabízejí dostatečné zabezpečení. Umožňují také autorizaci uživatelů, kterou je však třeba nejprve implementovat. Pokud jde o podporu dat, která jsou v systému posílána, nenabízí však žádný z nich žádné standardizované řešení, což se jeví jako nevýhoda oproti protokolu XMPP.

XMPP nabízí řadu rozšíření, která mají za úkol standardizovat některé často implementované funkce. Jedním z těchto rozšíření je GeoLocation, vedené pod označením XEP-0080. To umožňuje publikovat na server vlastní geografickou pozici, která je následně distribuována všem uživatelům, kteří si o ni zažádali. To je v kombinaci s již existující a fungující

sítí serverů, které tuto funkcionalitu implementují pro vyvíjený software, značnou výhodou oproti ostatním využitelným protokolům. Servery navíc již v základu podporují šifrování a autentizaci uživatelů a tu tak již není nutné implementovat v rámci systému. Protokol udržuje spojení s klientem a je tedy stavový. XMPP je navíc komunikační protokol, takže poskytuje již hotové řešení pro zasílání zpráv, což značně zjednodušuje implementaci. Přesto je ale nutné vyřešit reprezentaci složitějších mapových dat, jako je trasa, nebo polygon, která jsou v systému zapotřebí, protože stávající rozšíření XEP-0080 umožňuje pouze publikování vlastní polohy. Zmíněná pozitiva nicméně činí z XMPP vhodný protokol. Z toho důvodu se následující podkapitola hlouběji věnuje jeho vlastnostem a architektuře.

4.6 XMPP

V této podkapitole je pojednáno o protokolu, který je nejčastěji využíván jako komunikační protokol pro instant messaging. Tímto protokolem je eXtensible Messaging and Presence Protocol, zvaný též Jabber [18, 3].

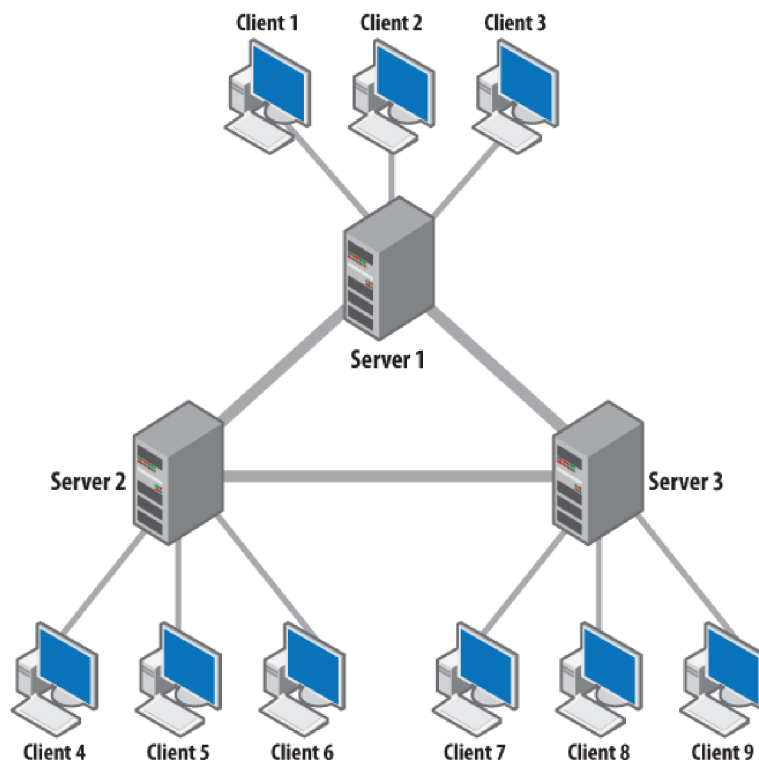
XMPP je otevřená, na XML založená technologie pro posílání zpráv mezi dvěma a více zařízeními v reálném čase. Jejimi hlavními vlastnostmi jsou decentralizovanost, rozšiřitelnost a bezpečnost. Od počátku byl navrhován jako protokol, který nebude v držení jednoho soukromého obchodního subjektu, což vede k tomu, že uživatelé nemusí při jeho používání podepisovat žádné smlouvy, platit nebo se obávat porušení licenčních pravidel. Tento fakt navíc usnadňuje možnost externích úprav ze strany vývojářské komunity.

Architektura XMPP sítě je složena z nezávislých serverů, z nichž žádný nemá dominantní roli. V podstatě lze architekturu XMPP přirovnat k elektronické poště, protože, jak lze vidět na obrázku 4.1, jsou jednotliví klienti připojeni k různým serverům a tyto servery jsou mezi sebou navzájem propojeny. Odeslání zprávy z jednoho klienta na jiného probíhá tak, že je zpráva nejprve doručena na server, odkud se, v případě, že se cílový klient nachází u jiného serveru, pošle na cílový server. Ten ji následně distribuuje požadovanému cíli. Navíc platí, že celý systém může přistupovat k jiným strukturám pomocí tzv. bran. Klient je identifikován pomocí jednoznačného identifikátoru JID, neboli Jabber Identifier, který má formát uživatel@server/zdroj. Uživatel je jméno, které bylo zvoleno při registraci, server je jméno serveru, na kterém je uživatel registrován a zdroj je volitelná položka rozlišující zařízení, ze kterého je momentálně přihlášen [18].

Spolu s tím, že je protokol otevřený a tím, že je primárně založený na formátu XML, přichází i jeho velmi snadná rozšiřitelnost. To vede k tomu, že v dnešní době existuje celá řada různých nadstavbových řešení, a k tomu, že je celá technologie neustále ve vývoji a dovoluje, aby si každý vývojář vybral ta rozšíření, která potřebuje a ta dále využíval při tvorbě svých vlastních implementací. Rozšíření jsou publikována v rámci tzv. XEP sérií¹⁴. Díky XML je protokol využíván nejen pro instant messaging, ale i pro standardní výměnu strukturovaných dat mezi programy. Využívá se tak například v sociálních sítích, nebo online hrách.

Jedním z existujících rozšíření je i XEP-0080 [4], který popisuje formát pro získávání geografických dat o uživateli. Jeho možná využití jsou publikování informací o pozici a získávání pozic jiných uživatelů. Při publikování pozice se využívá druhého rozšíření s názvem Publish - Subscribe, značeného jako XEP-0060. Ten slouží k registraci odběru určitého druhu informace od jiného uživatele, v tomto případě jeho geografické pozice. Toho se využívá tak, že není nutné formou multicastu zasílat změnu polohy všem ostatním uživatelům,

¹⁴XEP je zkratka pro XMPP Extension Protocol



Obrázek 4.1: Architektura XMPP sítě [18]

ale stačí tuto změnu poslat na XMPP server. Ten si udržuje seznam uživatelů, kteří jsou k odběru pozice od daného uživatele přihlášení a těm následně změnu distribuuje. To omezuje režii, která je nutná pro šíření informací o poloze mezi uživateli a celou proceduru tak zefektivňuje a z hlediska implementace značně zjednodušuje.

Geolokační data jsou uložena v elementu `geoloc` tak, jak je ukázáno v příkladu 4.4. Ten může obsahovat velké množství dalších subelementů, které jsou popsány v [4]. V popisovaném systému lze využít `lat`¹⁵ pro zeměpisnou šířku (řádek 7 v příkladu), `lon`¹⁶ pro zeměpisnou délku (řádek 8 v příkladu) a `accuracy` pro přesnost s jakou byla poloha získána.

```

1 <iq type='set' from='Novak123456@jabber.cz/n900'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
3   <publish node='http://jabber.org/protocol/geoloc'>
     <item>
5       <geoloc xmlns='http://jabber.org/protocol/geoloc'>
         <accuracy>20</accuracy>
7         <lat>49.817</lat>
         <lon>15.472</lon>
9       </geoloc>
     </item>
11  </publish>
13 </pubsub>
</iq>

```

Příklad 4.4: Publikování polohy pomocí XMPP

¹⁵latutude

¹⁶longitude

Zabezpečení protokolu částečně vyplývá z jeho decentralizace, protože vzhledem k tomu, že neexistuje jeden konkrétní páteří server, je prakticky nemožné ohrozit funkčnost sítě pomocí DOS útoku¹⁷. Kromě toho XMPP podporuje protokoly TLS a SASL. První pomocí šifrování XML streamů zajišťuje integritu a důvěrnost dat. Druhý realizuje spolehlivé ověření identity. Ta je zajištěna na vyšší úrovni než například u emailu, protože se uživatelé musí autentizovat u jejich serveru a zprávy od těchto uživatelů nemohou být podvrženy prostou změnou hlavičky díky šifrování zpráv. Další kontrola identity může být dosažena pomocí požadavku na potvrzení identity uživatelů na základě bezpečnostních certifikátů [18]. Oba typy zabezpečení jsou sice volitelné, ale servery mohou být konfigurovány tak, aby přijímaly pouze šifrovaná spojení. Jakmile je jednou nastaveno, že bude spojení šifrováno výše uvedenými metodami, je toto už šifrováno po celou dobu komunikace [26].

Jak již bylo zmíněno, je XMPP založen na výměně XML dokumentů. Pro pochopení mechanismu komunikace v systému je nejprve nutné definovat strukturu protokolu XMPP. Informace jsou převzaty z [18] a [3], kde je vše popsáno podrobněji. Komunikace začíná připojením klienta k serveru. Poté klient pošle hlavičku s označením verze XML, která bude v komunikaci používána. Následně je přenesen otevírací element stream, za kterým následuje celá další komunikace. Jakmile komunikace končí, je jako poslední přenesen uzavírací element stream, čímž je celá komunikace ukončena. Mezi dvěma hraničními elementy stream jsou přenášeny dokumenty s označením stanza, které lze přirovnat k jednotlivým paketům. Stanzu lze rozdělit na 3 typy.

Message je hlavním prostředkem výměny informací mezi dvěma účastníky komunikace. Rozšíření XEP-0045 navíc přidává i možnost komunikace více účastníků. Každá zpráva obsahuje atribut type, který rozlišuje, o jaký typ zprávy se jedná. Dalšími atributy jsou from a to, které označují odesílatele a příjemce zprávy a volitelné id, které identifikuje konkrétní zprávu.

Presence je prostředkem pro získávání informací o dostupnosti daného uživatele. Jeho užívání je omezeno tím, zda má hledaný uživatel hledajícího uživatele ve svém seznamu kontaktů. Nejčastějším typem notifikace je stav online, nebo offline. Definován je však i nepovinný element show, který obsahuje informaci o částečné dostupnosti uživatele. Dalším volitelným elementem je status, který může obsahovat uživatelem vytvořenou informaci o jeho stavu. Element priority značí prioritu zprávy v rozsahu -127 až 128. Implicitní hodnotou je 0. Poslední element, error, obsahuje informaci o chybě. Kromě toho obsahuje presence, stejně jako message, atribut type, značící její typ.

Posledním typem stanzy je IQ, což je zkratka pro Info/Query. Tento druh stanzy je založen na principu dotaz - odpověď, které se párují díky atributu id, který je identifikuje. Dotaz a odpověď může nabývat typů get, set, result a error. IQ se používá kupříkladu ke zjišťování informací o uživateli, zasílání chybových hlášení a zkoumání, jaké služby druhá strana poskytuje. Pomocí IQ se také spravuje seznam uživatelů. Ten je vždy uložen na serveru, takže nezáleží na tom, z jakého zařízení se uživatel přihlásí. Vždy bude mít k dispozici seznam všech svých kontaktů [18].

¹⁷Denial Of Service

Kapitola 5

Návrh

V této kapitole je představen návrh projektu. V první části jsou probírány jednotlivé případy užití, je diskutováno, jací uživatelé s výsledným systémem mají pracovat a jaké jsou jejich role. Dalším bodem kapitoly je návrh architektury z hlediska komponent, z nichž se systém skládá, a jak jsou tyto technicky realizovány. Při psaní této kapitoly bylo čerpáno z [20].

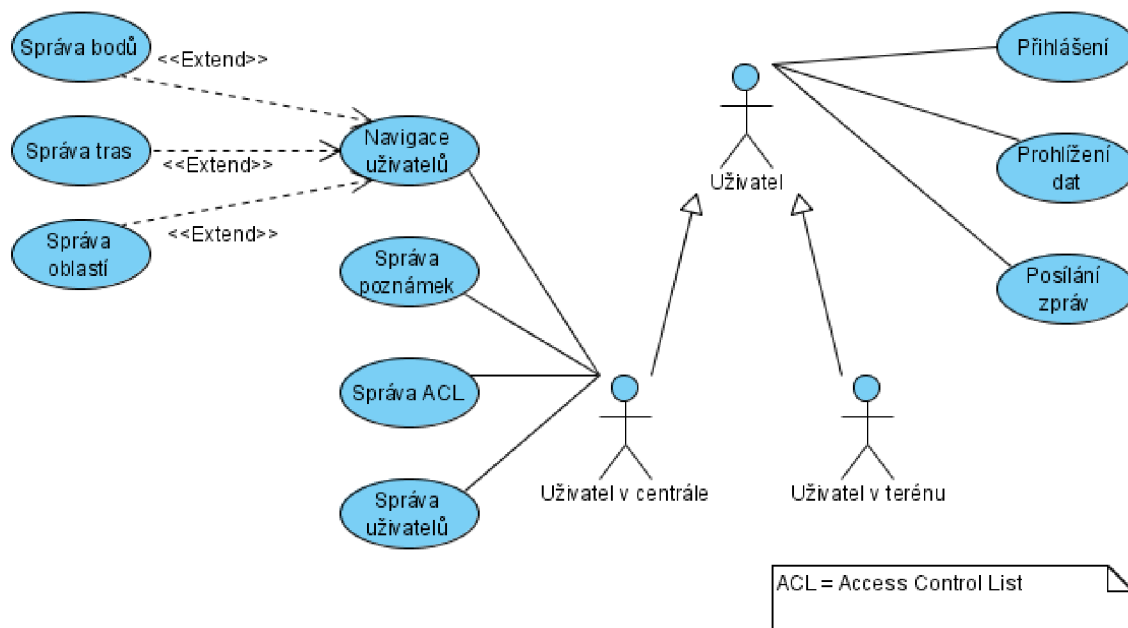
5.1 Případy užití

Systém je primárně zaměřen na řízení lidí a jejich činností v terénu. Proto je podstatné, jakým způsobem jsou rozděleny role jeho uživatelů a k jakým činnostem má uživatel přístup, či jaké informace mu systém poskytuje. Diagram 5.1 zobrazuje případy užití spojené s dále uvedenými aktéry.

V systému se vyskytují 2 typy uživatelů. Jednak je to uživatel nacházející se v centrále, neboli koordinátor. Druhým uživatelem je uživatel v terénu, neboli jednotka. Koordinátor má za úkol sledovat pohyb jednotek po mapě a vyhodnocovat z něj a z okamžité situace, jakými cestami se ta která jednotka může dostat na místo určení. K tomu, aby se na místo určení dostala, vytváří koordinátor na mapě 3 druhy grafických objektů. Jednak je to bod, který slouží pro označení jednoho konkrétního místa. Dalším je čára, reprezentující trasu, po které se má jednotka vydat, a třetím je polygon, který slouží k ohraničení určité oblasti. K tomu, aby každá jednotka viděla pouze relevantní informace, vytváří koordinátor seznamy uživatelů, kteří mají k dané informaci přístup, tzv. access control list (dále jen ACL). Přiřazováním vytvářených mapových objektů k těmto seznamům je realizována klasifikace informací. Koordinátor může také vytvářet nové uživatele.

Druhým typem uživatele je jednotka. Ta se po zapnutí přístroje automaticky přihlásí do systému a poté sleduje na svém mobilním zařízení informace, jež jí vytvořil koordinátor, přičemž nevidí informace všechny, ale pouze ty, ke kterým má v rámci svého ACL přístup. Navíc může sledovat pohyb ostatních jednotek, které se nacházejí v její blízkosti, a to buď všech aktuálně přihlášených do systému, nebo těch, se kterými je zařazena do stejného ACL. Kromě toho mohou oba zmínění uživatelé komunikovat formou jednoduchého chatu.

Z neformální specifikace plynou následující požadavky na systém: Abstraktním aktérem je Uživatel. To je nejobecnější aktér komunikující s vybranými akcemi, kterými jsou prohlížení dat v systému a komunikace pomocí zaslání zpráv. Konkrétním potomkem je Jednotka. Speciálním akterem je koordinátor, což je aktér s možností prohlížet všechna data, vytvářet, editovat a mazat všechny mapové objekty a access control listy. Tento aktér dále může spravovat jednotky, posílat zprávy a vytvářet, editovat a mazat poznámky. Vyu-



Obrázek 5.1: Diagram případů užití

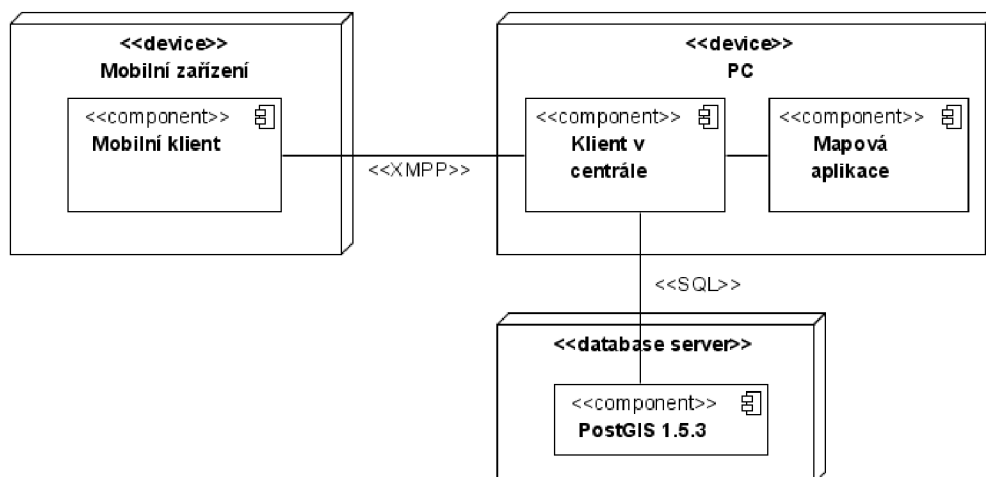
žití dědičnosti aktérů vyplynulo ze společných požadavků (případů užití) aktérů v různých rolích.

5.2 Architektura

Vzhledem k cíli práce, realizaci programu pro efektivní řízení jednotek v terénu, musí být výsledný systém sestaven z několika různých komponent. Proto je nutné definovat nejen strukturu každé z nich, ale i způsob, jakým spolu navzájem komunikují. Výsledný systém v sobě musí zahrnovat dvě různé složky. Jednak jsou to mobilní telefony, jichž využívají jednotky v terénu a jednak je to stanoviště v centrále. Systém musí také zahrnovat datové úložiště pro perzistentní uchování a analýzu dat. Výsledná architektura je znázorněna na diagramu 5.2. V následujících odstavcích jsou prvky jeden po druhém rozebrány.

Mobilní klient je fyzicky reprezentován mobilním telefonem, na němž běží klientská mobilní aplikace. Ta se stará o zobrazování dat na mapě, stejně jako o interakci uživatele s okolím a sběrem dat o jeho poloze a jejich odesíláním na centrálu. Uživatelské rozhraní ponechává největší část obrazovky mobilního telefonu pro mapu. Mapa umožňuje vykreslení 3 druhů dlaždic, kterými jsou satelitní zobrazení, terénní pohled a klasické zobrazení. Při pohybu v terénu je možné nastavit, zda bude mapa sledovat pohyb uživatele a uchovávat jeho pozici vždy uprostřed obrazovky. V rámci aplikace je možné po mapě pohybovat a sledovat koordinátorem vytvořené grafické objekty a pozice další jednotek. Kromě toho se u grafických objektů mohou nacházet i doplňující informace vytvořené v centrále ve formě poznámek. Ty slouží pro upřesnění významu daného objektu. Po dotyku prstu v místě poznámky se objeví panel s textem poznámky. Z aplikace lze rovněž poslat zprávu na centrálu. Klientská mobilní aplikace nevyužívá explicitně vlastní lokální databázi. Tu částečně nahrazuje uložení mapových objektů ve formátu `QLandmark`, který zajišťuje perzistenci i po vypnutí aplikace.

Klient na straně koordinátora je realizován jako desktopová aplikace s uživatelským



Obrázek 5.2: Architektura systému

rozhraním. V rámci něj lze vytvářet jak jednotky, tak i access control listy. Po vytvoření ACL je možné vybírat ze seznamu jednotky a ty k němu přiřazovat. Takto přiřazená jednotka je následně označena jako obsazená. Hlavním prvkem uživatelského rozhraní je mapa, jejíž návrh je na obrázku 5.3. Ta slouží koordinátorovi k vytváření mapových objektů. Ty jsou vždy přiřazovány k vybranému ACL a distribuovány jednotkám, které k tomuto ACL byly přiřazeny. Jednotkám je také možné posílat zprávy. Poslední funkcí, kterou může koordinátor využívat, je analýza historie. Veškeré změny, které jsou v uživatelském rozhraní provedeny, jsou ukládány do databáze. Kromě toho klient na pozadí na jedné straně sbírá informace o pozici mobilních klientů, které ukládá do databáze a zobrazuje na mapě. Na straně druhé pak periodicky kontroluje stav databáze a sleduje, zda nevypršel čas platnosti některého ACL. V kladném případě je tento ACL označen v databázi jako neplatný.

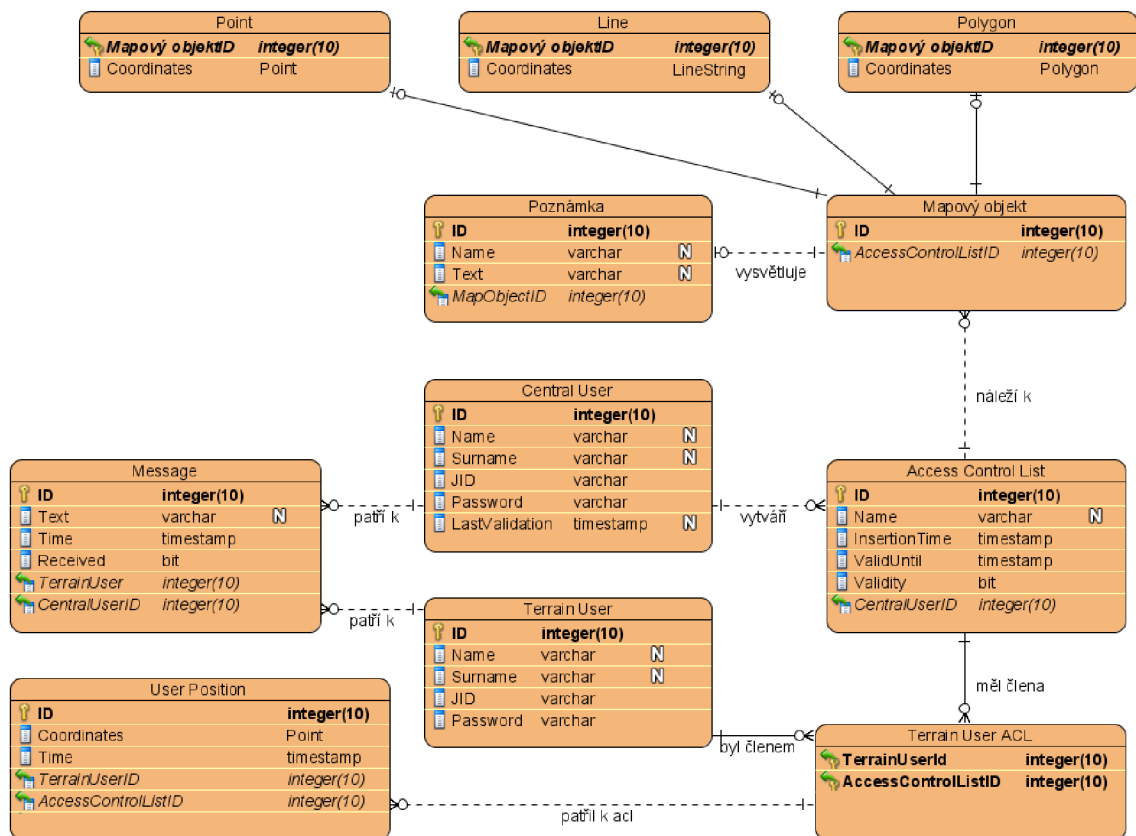


Obrázek 5.3: Návrh uživatelského rozhraní

Databáze je realizována s využitím databázového systému PostgreSQL s rozšířením PostGIS ve verzi 1.5.3. Na diagramu 5.4 je vidět návrh jejího schématu. V databázi se ukládají uživatelé, ACL a data, s nimiž uživatelé pracují. Každý uživatel je unikátně identifikován svým šestimístným Osobním evidenčním číslem, což je unikátní identifikační číslo každého příslušníka Policie ČR. Základní tabulkou v databázi je Mapový objekt, který reprezentuje buď bod, nebo trasu, nebo polygon. Podle toho, o jaký typ objektu se jedná, je jeho geometrie zaznamenána buď jako geografický typ Point, LineString, nebo Polygon. Využití geografických datových typů umožňuje realizovat dotazy nad databází za účelem ověření, zda se jedotka nachází v prostoru, který jí byl přidělen k prohledávání, jak dlouhá je trasa, kterou musí jednotka absolvovat, nebo jak velkou plochu zabírá prohledávaná oblast. Ke každému mapovému objektu může být v databázi uložena poznámka, která rozšiřuje jeho význam.

Důležitou tabulkou v databázi je access control list, který spojuje uživatele s geografickými objekty, které pro něj byly vytvořeny. ACL mohou mít přiřazenu dobu platnosti, nebo může být jejich platnost ukončena manuálně. Z toho důvodu je jedním z jejich atributů validita, která je implicitně nastavena na hodnotu true a v okamžiku zneplatnění se mění na false. Z platnosti ACL vyplývá následně také platnost všech geografických objektů, které k němu náležejí.

Podobným způsobem jako Bod se v databázi ukládá i aktuální pozice náležející ke konkrétní jednotce. Datovým typem pozice je typ Point, ukládaný spolu s časem, kdy byla přijata. Protože uživatel může po dobu své existence náležet do více než jednoho ACL, existuje v databázi pomocná tabulka **Terrain User ACL**, která tuto informaci uchovává, aby mohla být využita zjištění do jakého ACL uživatel patřil v době, kdy byla přijata jeho geografická pozice. To umožňuje analyzovat zpětně jeho pohyb a vybírat pouze ty souřadnice, které jsou pro daný časový interval podstatné. V databázi se též zaznamenávají zprávy, které si uživatelé mezi sebou poslali.



Obrázek 5.4: Návrh schématu databáze

Kapitola 6

Implementace

Tato kapitola se věnuje popisu implementace vyvinutého systému. V první části se zaměřuje na představení a popis realizace funkcionality, která byla v projektu implementována. V další části tuto funkcionality zasazuje do kontextu uživatelského rozhraní. Popisuje, jakým způsobem lze představené funkce vyvolat a jaký vizuální výstup uživateli zprostředkovávají. Závěr kapitoly se věnuje implementačním problémům, které při vývoji nastaly a popisuje, jakým způsobem byly vyřešeny. Při psaní této kapitoly bylo využíváno [12].

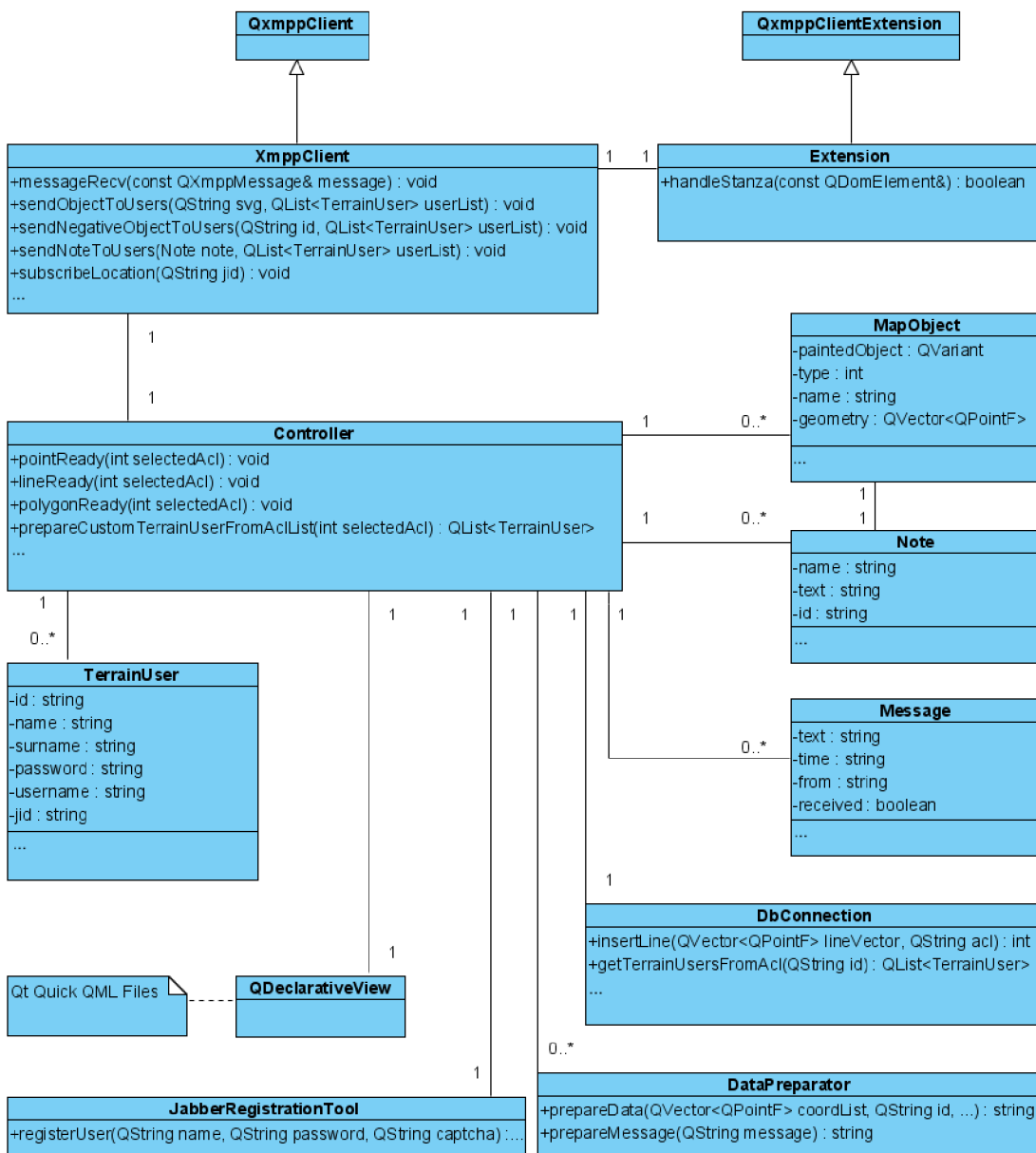
6.1 Funkcionalita

Tato podkapitola popisuje způsob, jakým je architektonický návrh z kapitoly 5 realizován v rámci koordinačního systému Policie České Republiky. Představeny jsou také praktické příklady, jak lze systém v tomto kontextu využít.

Společnými prvky části na straně koordinátora, jejíž zjednodušený diagram tříd je vidět na obrázku 6.1, a klientské části znázorněné na diagramu tříd na obrázku 6.2, jsou dva. Jednak je to mapa, na níž jsou na straně koordinátora zakreslovány grafické objekty a na straně jednotky tyto objekty zobrazovány. Druhým prvkem je komunikační protokol XMPP, kterým jsou grafické objekty posílány.

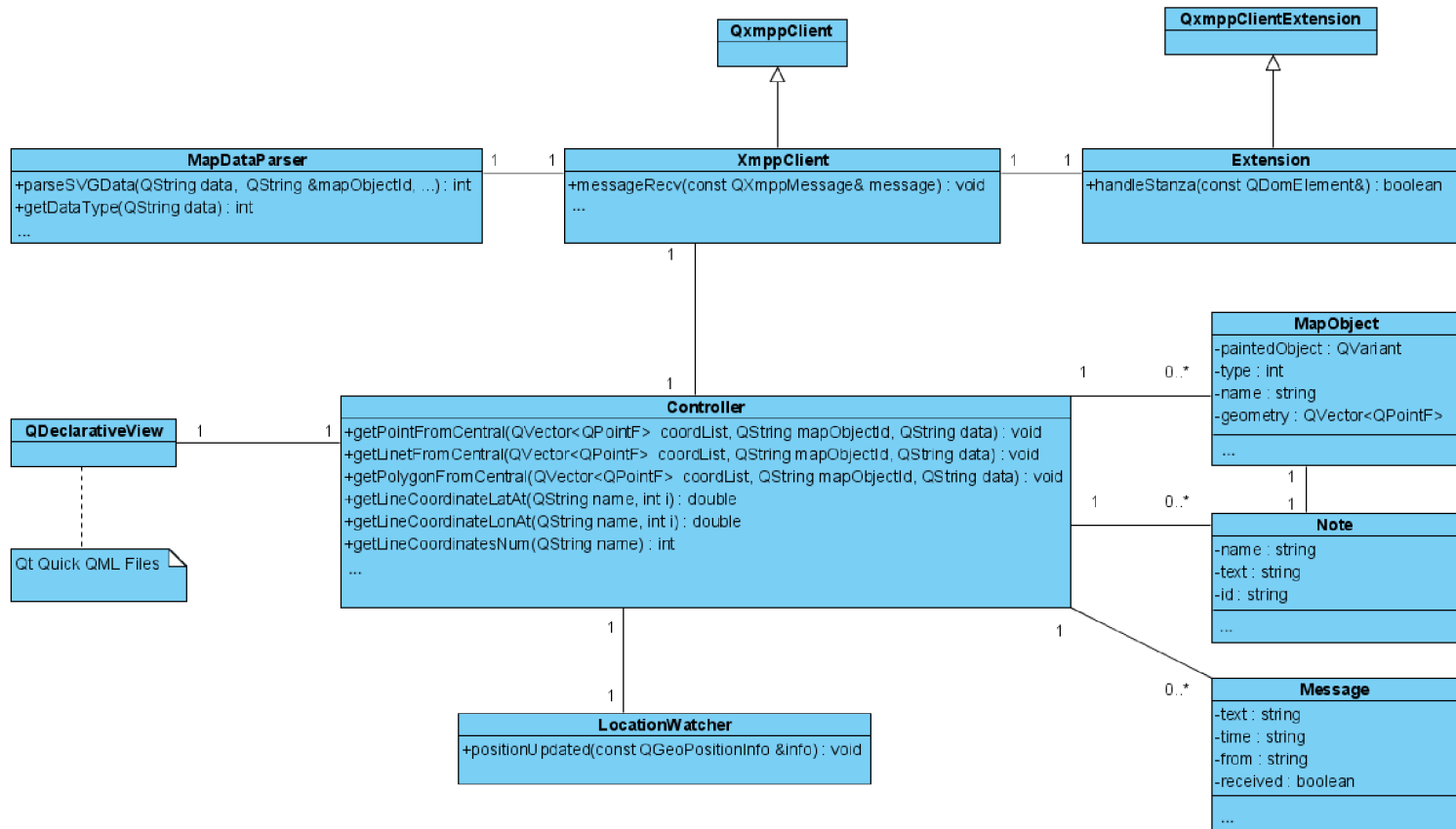
Mapa, jak je možné vidět na obrázku 6.5, umožňuje zobrazovat tři typy grafických objektů. Prvním je bod, který se může vyskytovat ve čtyřech barevných provedeních. Zelená označuje bod, který do mapy zanesl koordinátor a reprezentuje určité místo, na které se má jednotka dostat, nebo které může sloužit pro vyznačení určitého významného orientačního bodu, jímž se může jednotka při své navigaci v terénu řídit. Modrá barva bodu znázorňuje vlastní pozici, nebo pozice ostatních jednotek, nacházejících se v okolí. K rozeznání jednotek je u každého modrého bodu uvedeno jméno jemu odpovídající jednotky. Bod označený šedě označuje jednotku, s níž bylo ztraceno spojení. Tímto způsobem se automaticky znázorňuje, že jednotka aktuálně neposílá svoji geografickou polohu a nemůže tak být považována za nehybnou. Žlutý bod se vyskytuje pouze na straně koordinátora a označuje bod, který byl koordinátorem vybrán pro úpravu.

Druhým grafickým objektem je čára. Ta může mít v současném stavu implementace pouze červenou barvu. Jejím účelem je vytyčení trasy, po níž se má jednotka dostat na místo určení, nebo vymezení, kudy se vydal například pronásledovaný pachatel. Čára se po kliknutí změní na žlutou, což značí její výběr za účelem editace. Posledním grafickým objektem je polygon vymežující určitou část prostoru, v níž má například probíhat pátrání, nebo jíž se mají jednotky z nějakého důvodu vyhnout. Polygon má na mapě modrou barvu,



Obrázek 6.1: Zjednodušený diagram tříd pro centrálu

Obrázek 6.2: Zjednodušený diagram tříd pro mobilního klienta



příčemž po označení se změní jeho hranice na žlutou.

Ke každému grafickému objektu lze po jeho vybrání koordinátorem nastavit popis. Ten má dvě složky. Jednak je to název, který se zobrazuje přímo pod daným objektem, a dále je to text popisu, jenž se zobrazí jednotce až po označení na název daného objektu prstem. Pro rozlišení, zda daný popis obsahuje ještě doplňující informace, slouží změna barvy názvu na modrou a jeho podtržení, které má evokovat hypertextový odkaz z HTML stránek. Z důvodu nedostatečných nástrojů pro vytváření grafických objektů na straně použitého mapového widgetu má popis podobu pouze jednoduchého textu. Pro jeho lepší zvýraznění je tento text podbarven bíle.

Systém umožňuje koordinátorovi vytvářet nové uživatele. K tomu je nutné zadat jeho osobní údaje, jako je jméno, příjmení a heslo a dále login, jež slouží pro vytvoření jabberového účtu na XMPP serveru. Současná implementace využívá služeb serveru Jabber.cz, který podporuje všechna rozšíření, která jsou systémem využívána. Daní za to je, že je při registraci uživatele nutné zadat kód CAPTCHA¹, což je test, ve kterém musí uživatel opsat text z obrázku, aby prokázal, že je člověkem a nikoli programem. Nutnost opisování textu spolu s časovým omezením registrace jednotlivých uživatelů je možné vynechat vytvořením vlastního XMPP serveru. Jakmile je uživatel vytvořen, je ihned na straně koordinátora paralelně s ním přihlášen k serveru a rozesílá žádosti o autorizaci všem aktuálně v databázi existujícím uživatelům. Ti automaticky po přihlášení tuto žádost potvrzují. V dalším kroku nově vytvořený uživatel posílá na server přihlášky k odběru polohy všech uživatelů tak, jak je definováno v rozšíření XEP-0060.

Využití standardizovaného řešení pro distribuci geografické pozice uživatele prostřednictvím rozšíření Geolocation v XEP-0080 má jak pozitiva, tak i negativa. Pozitivem je, že systém může využít i osoba, která nemá k dispozici implementovaného klienta. Toho lze na systémech Maemo a MeeGo docílit tak, že využije například aplikaci Azimuth². V takovém případě sice nebude schopna přijímat žádná specifická data, jako jsou například grafické objekty, zato ale bude moci systému poskytovat vlastní polohu, což pro účely Policie ČR může mít užitek například v případě, že byla do systému zařazena osoba, která jej nesmí z nějakých důvodů využívat, ale jejíž polohu je účelné sledovat. Nevýhodou v předchozím odstavci zmíněného řešení, kdy klienti automaticky potvrzují žádosti o autorizaci je, že si je tak může do svého rosteru zařadit i externí uživatel. Tomu sice nebudou zasílány grafické objekty a zprávy, protože není v databázi, bude ale v případě implementace vlastního mapového klienta schopen sledovat pozice uživatelů v systému. Tento fakt lze nicméně eliminovat dvěma způsoby. Tím jednodušším je zrušit automatické potvrzování autorizace a veškeré autorizace provést ručně například s využitím již existujících jabberových klientů. Složitějším způsobem je vytvoření vlastního XMPP serveru, u něhož je nastaven zákaz komunikace s jinými servery v síti. Takovýto server pak neodpovídá uživatelům zaregistrovaným u jiných serverů a není tedy možné si jeho uživatele přidávat do rosteru. Podobné řešení využívá například Facebook [27].

Access control listy, popsané v podkapitole 5.2 jsou v rámci implementace pro Policii ČR interpretovány jako akce, a takto jsou i v uživatelském rozhraní značeny. Každá akce má vlastní název, přiřazené jednotky a vytvořené objekty. Příkladem pro ilustraci významu akce může být situace, kdy koordinátorovi volá občan, že mu někdo vykrádá dům. Koordinátor v takovém případě vytváří novou akci, které zadá jméno a volitelnou dobu platnosti a přiřazuje k ní jednotky, jež má aktuálně k dispozici. Po vytvoření akce a jejím zvolení v pohledu s mapou jsou následně všechny do mapy zakreslované objekty v reálném čase

¹Completely Automated Public Turing test to tell Computers and Humans Apart

²<http://maemo.org/packages/view/azimuth/>

distribučovaný na mobilní zařízení těch jednotek, které k dané akci náleží.

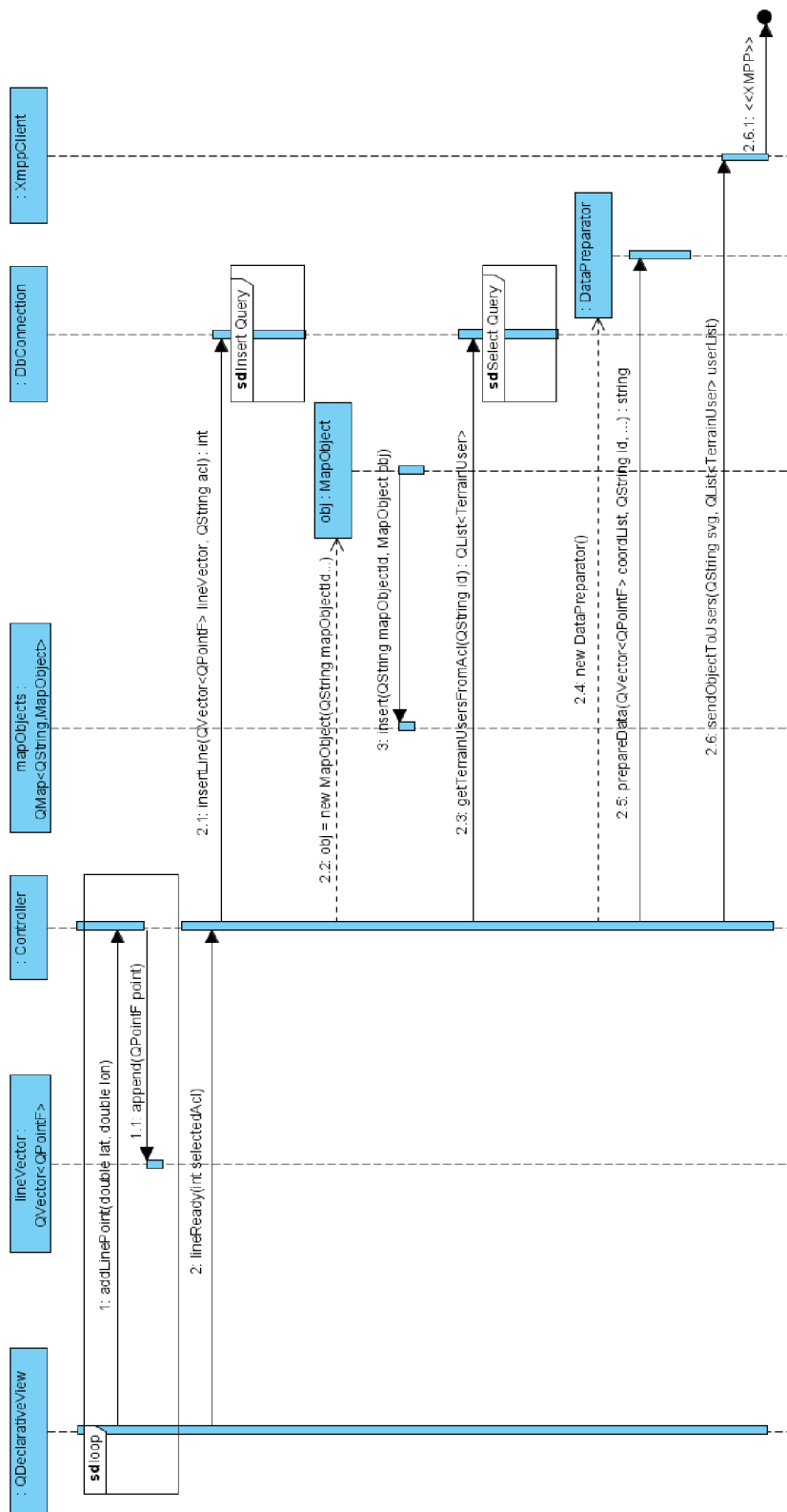
Jak lze vidět na diagramu 6.3, je při kreslení nového mapového objektu vytvářen vektor bodů, které popisují jeho geometrii. Po dokončení tvorby je objekt nejprve uložen do databáze a následně do asociativního pole, díky čemuž je možné s ním v rámci uživatelského rozhraní dále manipulovat. Poté jsou z databáze získáni uživatelé, kterým má být objekt distribuován. Ještě před odesláním těmto uživatelům jsou souřadnice objektu transformovány do formátu SVG. Takto reprezentovaný vektorový obrázek je následně přenesen v těle XMPP zprávy každému klientovi z dané akce.

Schéma kroků po přijetí grafického objektu u klienta je znázorněno na diagramu sekvence na obrázku 6.4. Klient z přijaté zprávy vezme její tělo a to analyzuje, aby zjistil, o jaký druh informace jde. Následně zprávu zpracovává, aby získal data, která transformuje do interní reprezentace. Tou je jak na straně jednotky, tak koordinátora, vektor bodů znázorňující geometrii daného objektu. Tento objekt je následně vykreslen na mapě.

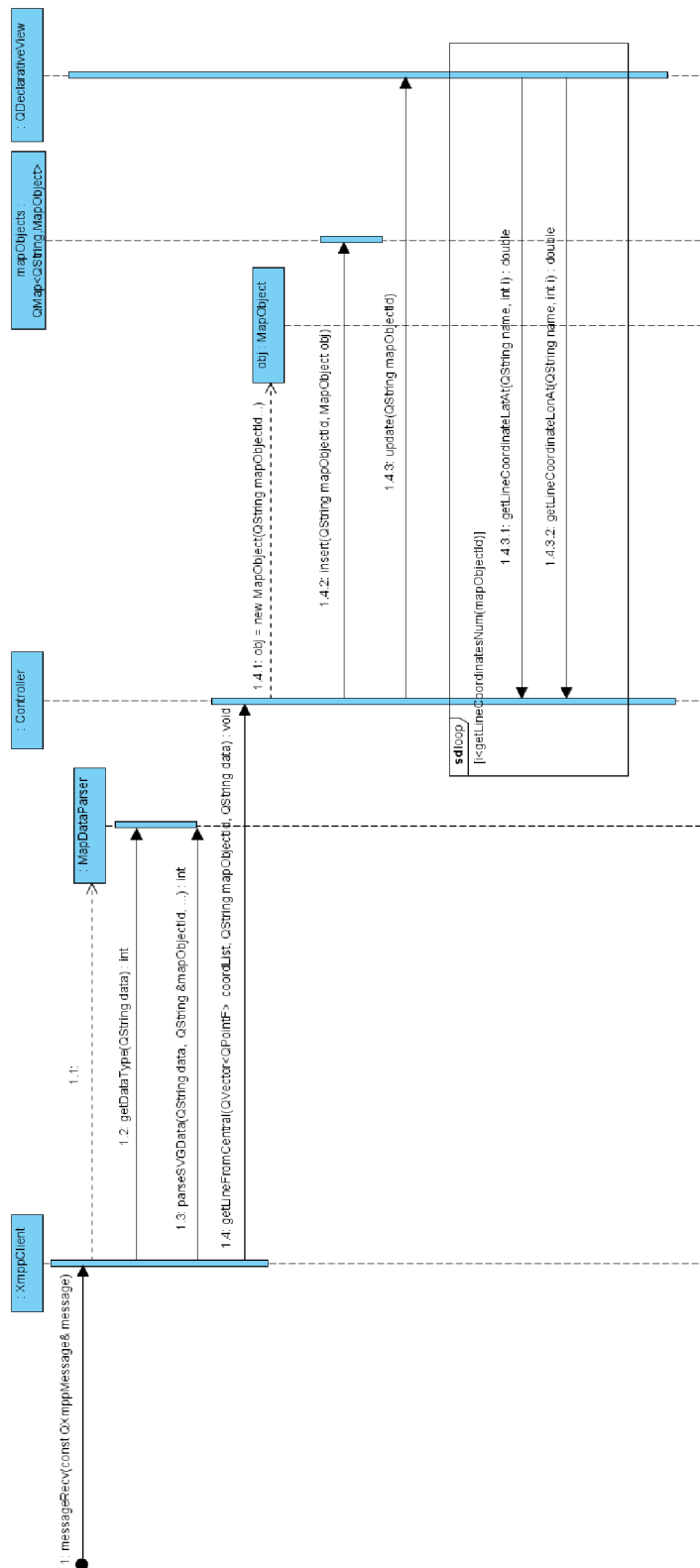
Každý zasílaný objekt má v rámci své SVG reprezentace nastaveno jméno, které odpovídá identifikátoru v databázi, jak lze také vidět na diagramu 6.4. Tím je zaručena jednoznačnost každého objektu, který je zobrazen na nějakém mobilním zařízení. Kromě bodu, čáry a polygonu systém zasílá také "negativní" objekt, což je zpráva obsahující pouze jméno určitého v databázi se vyskytujícího objektu. Po obdržení této zprávy klient vyhledá grafický objekt s daným jménem a tento objekt odstraní z uživatelského rozhraní. Toho se využívá, když je ukončena platnost určité akce, což se může stát buď automaticky po uplynutí doby její platnosti, nebo explicitně po zadání koordinátorem. V takovém případě je akce v databázi učiněna neplatnou a jsou vybrány všechny do ní náležející grafické objekty. Ty jsou odstraněny z koordinátorova uživatelského rozhraní a jejich identifikátory rozeslány odpovídajícím uživatelům, jak je popsáno výše.

Posledním typem zasílané informace je zpráva, která slouží pro komunikaci mezi koordinátorem a jednotkou. Využita může být pro jakékoli předání informace, které se z nějakého důvodu nehodí pro přenos vysílačkou, nebo pro doplnění vizuálních informací zobrazených na mapě. Typickým příkladem může být SPZ hledaného automobilu, nebo přesná adresa domu, jejichž diktování hlasovou formou není na rozdíl od jasné textové reprezentace efektivní. Jednotkám není dovoleno zasílání zpráv mezi sebou. Tato funkcionality je principiálně podporována, není však implementována z toho důvodu, že takovéto chování není podle specifikace zástupců Policie ČR vyžadována. V případě její dodatečné implementace je možné posílat zprávy přímo mezi jednotkami bez vědomí centrály, což je z hlediska evidence nežádoucí. Vhodnější možností je tak každou zprávu poslat na centrálu a z ní ji přeposlat odpovídajícímu adresátovi.

Všechna data, jako jsou grafické objekty, zprávy, uživatelé, akce a pozice uživatelů jsou uchovávána v databázi za účelem další analýzy. Ta má pro Policii ČR zásadní význam, protože podle ní může zjistit, jakým způsobem plnila určitá jednotka jí přidělené úkoly. Lze tak sledovat kteří uživatelé byli k dané akci přiřazeni, jaká komunikace mezi nimi a centrálou probíhala a kudy se pohybovali. Lze také vykreslit do mapy všechny objekty, které byly v rámci dané akce vytvořeny a přes ně vykreslit trasu jednoho konkrétního uživatele, nebo všech uživatelů z této akce. Toho lze využít například v modelovém případě, kdy je úkolem prohledat například les z důvodu nalezení pohřešované osoby. Vymezením oblasti pomocí polygonu lze určit oblast, která se má prohledávat, a spustit pátrání. Pohyb jednotek lze v reálném čase sledovat a jejich trajektorie případně upravovat. Pokud hledání dopadne neúspěšně, přichází na řadu analýza, kdy jsou všechny grafické objekty včetně polygonu opět vykresleny do mapy a je znázorněn pohyb všech jednotek. Podle policejních tabulek má člověk procházející lesem zhruba desetimetrový výhled na obě strany, čemuž odpovídá



Obrázek 6.3: Vytvoření a odeslání trasy z centrály



Obrázek 6.4: Přijetí a zobrazení trasy na mobilním klientovi

tloušťka čáry, jež se vykresluje pro znázornění pohybu jednotky. Studium volných míst, která zůstanou po takto vykreslených trasách lze odhalit, kde by se zmizelá osoba mohla potenciálně nalézat. Následně lze vytvořit novou akci a na tato místa vyslat nové jednotky a případně tam zmizelou osobu nalézt.

6.2 Popis uživatelského rozhraní

Tato podkapitola popisuje uživatelské rozhraní jak na straně koordinátora, tak i na straně mobilního klienta. Jejím smyslem je seznámit čtenáře s realizací uživatelského rozhraní, rozmístěním ovládacích prvků, a specifickými vlastnostmi uživatelského rozhraní vzhledem k použitým platformám, jak jsou popsány v podkapitole 6.1.

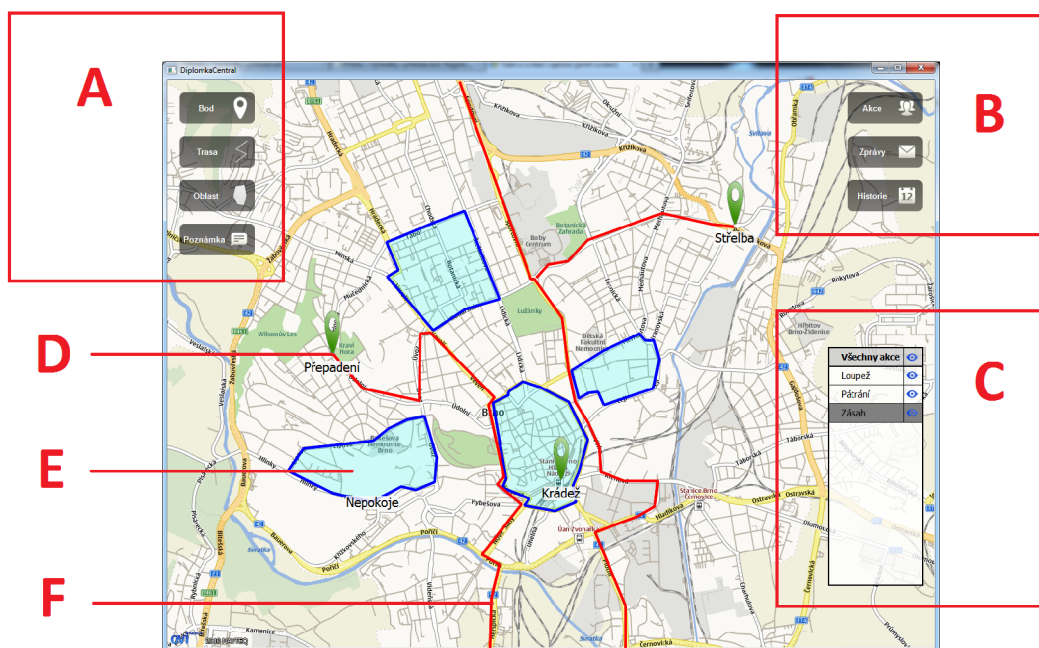
Hlavní částí uživatelského rozhraní na centrále je mapa. Na té se zobrazují grafické objekty bod, čára a polygon, jak lze vidět na obrázku 6.5 v částech D, E a F. Každý z těchto objektů reaguje na kliknutí změnou barvy na žlutou, což značí že lze objekt buď smazat, nebo mu přiřadit popis. V části A jsou 4 tlačítka, z nichž první 3 po kliknutí zůstávají označena. Jejich účelem je přepínání módů pro vytváření grafických objektů. Po stisknutí tlačítka Bod tak uživatel může vytvořit, po kliknutí do mapy na daném místě, bod. V případě, že vytváří čáru nebo polygon, je tento objekt vytvářen vícenásobným klikáním do mapy, přičemž tvar výsledného objektu je znázorněn tenkou oranžovou čarou. Jakmile poté uživatel stiskne pravé tlačítko myši, objekt se vytvoří, zobrazí se na mapě a odešle se uživateli z právě označené akce. Čtvrté tlačítko lze zvolit až v momentě, kdy již je označen některý grafický objekt. V takovém případě se objeví okno, v němž je možné vybranému objektu přiřadit popis, jenž se skládá ze jména a vlastního textu.

V části B jsou 3 tlačítka. První slouží pro vyvolání okna v němž je možné vytvářet uživatele a akce. Obojí je zobrazeno v seznámech, přičemž označením akce a uživatele je možné tohoto uživatele k dané akci přiřadit. Uživatele i akce je možné smazat. Akce lze navíc označit jako ukončené, čímž se odstraní z uživatelského rozhraní, ale jsou nadále dostupné v sekci Historie.

Druhým tlačítkem jsou Zprávy. Ty vyvolají okno, v němž je možné vybírat uživatele a zobrazovat zprávy přijaté a odeslané zprávy. Zvolenému uživateli lze také zprávu poslat. V případě, že koordinátor obdrží novou zprávu a nachází se mimo okno se zprávami, zabarví se tlačítko Zprávy žlutě. V okně se zprávami je následně žlutě označen uživatel, od kterého byla nová zpráva obdržena.

Třetím tlačítkem v popisované části je tlačítko Historie. To vede na obrazovku, v níž lze zvolit interval, v němž se mají vyhledat akce, které v něm začaly a skončily. Po označení určité akce jsou zobrazeny jednotky, které do ní patřily a označováním jednotek lze vidět, jaké zprávy byly mezi ní a centrálou poslány. Kliknutím na tlačítko Zobrazit se uživatelské rozhraní přepne na mapu, do níž jsou zakresleny všechny mapové objekty, které byly v rámci dané akce vytvořeny. Ty již není možné označovat. V tomto pohledu na mapu lze následně přepínat mezi akcemi z vybraného časového intervalu a volbou jednotek ze zobrazeného seznamu do mapy vykreslovat trajektorie jejich pohybu. V části C jsou vypsány všechny momentálně platné akce. Jejich výběrem je možné mezi nimi přepínat a tak volit, pro kterou akci budou přiřazovány a rozesílány právě vytvářené grafické objekty. Volbou modré ikony se na mapě zobrazí pouze ty objekty, které patří k dané akci. Kliknutím na tuto ikonu u nejhornějšího řádku seznamu označeného jako Všechny akce se do mapy opět vykreslí objekty ke všem akcím.

Grafické uživatelské rozhraní mobilních klientů je podobné, jako uživatelské rozhraní koordinátora. Jeho znázornění lze vidět na obrázku 6.6. Zde se v části 1 nachází tlačítko. To



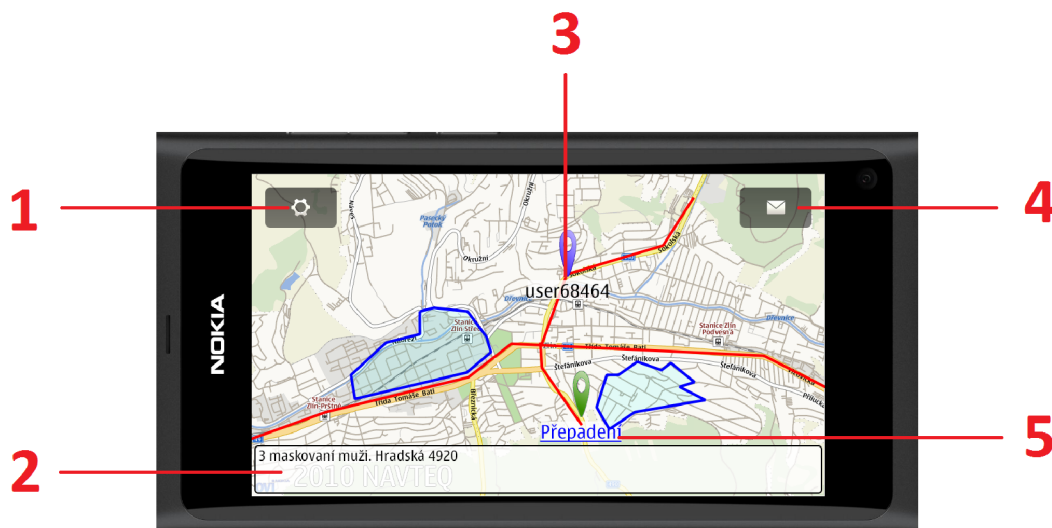
Obrázek 6.5: Uživatelské rozhraní v centrále

po stisknutí vyvolá obrazovku, v níž se lze přihlásit jako jiný uživatel, nebo vypnout aplikaci. V části 4 je další tlačítko, které vede na obrazovku, v níž lze číst zprávy od koordinátora a odpovídat na ně. Část 3 označuje bod, který ukazuje pozici aktuálního uživatele. Stisknutím modře obarveného a podtrženého textu v části 5 u mapového objektu je možné zobrazit text popisu, který náleží k danému mapovému objektu. Tento text se zobrazí ve vlastním panelu, který je možné vidět v části 2. Tímto panelem je možné posouvat, takže umožňují zobrazit větší množství textu.

6.3 Problémy při implementaci

V průběhu vývoje vyšla na povrch řada nedostatků, chyb a obtíží, jež bylo nutné řešit. Tato podkapitola podává výčet těch největších a každý problém rozebírá z hlediska jeho dopadu na výslednou aplikaci a popisuje, jak byl vyřešen.

Systém využívá část knihovny Qt Mobility, která má název Landmarks [38]. Jedná se o API pro práci s geografickými značkami. Toto API lze využít pro perzistentní uchování bodů, jejich filtrování a také vykreslování do mapy. Všechny tyto vlastnosti jsou v systému použity pro rychlou správu všech grafických objektů v rámci uživatelského rozhraní. Složitější geometrie pro trasu a polygon jsou uchovávány v asociativním poli, odkud jsou při zobrazování jim odpovídajícího značek získávány. Značky jsou spravovány manažerem značek, který by měl být podle dokumentace schopen zaznamenat změny, které se v jím spravovaných značkách vyskytnou. To ale bohužel platí pouze pro přidání nové značky, kdy je tato správně vykreslena do mapy, i při jejím smazání, kdy je z mapy odebrána. Nefunguje však tehdy, když je některá ze značek pouze upravena. V takovém případě se její atributy správně změní, ale s vykresleným objektem už se nic nestane. To je nevhodné například při změnách pozice uživatelů, k nimž dochází poměrně často. Řešením mohlo být smazání existující značky z manažeru a její opětovné přidání. To však nejde a změna se projeví až



Obrázek 6.6: Uživatelské rozhraní na mobilním telefonu

tehdy, když je po ní přidána nějaká další značka. Řešením v rámci aplikace je uchování reference přímo na v mapě vykreslený mapový element v rámci zmíněného asociativního pole, kdy změna parametrů tohoto elementu vyvolává i změny v jeho vzhledu.

Druhým problémem, který souvisí s uživatelským rozhraním a vykreslováním objektů je implementační chyba ve třídě grafické reprezentace mapového polygonu, která je taktéž součástí Qt Mobility. Jak bylo zjištěno kontrolou zdrojových kódů Qt Mobility, vychází chyba ze špatně napsaného validátoru daného objektu, který má kontrolovat, že se v deklaraci objektu uživatelem nevyskytují jiné elementy, než ty, které mu náleží. Závada spočívá v tom, že validátor, na rozdíl například od stejně realizované čáry, nebere jako validní žádné elementy kromě elementu Coordinate, který reprezentuje jeden bod geometrie polygonu, a tak do něj nedovoluje přidat například element pro odhycení události kliknutí myši. Tato chyba byla vyřešena vnořením elementu polygonu do jiného elementu MapGroup, kterému byly požadované vlastnosti nadeklarovány.

S předchozím řešením souvisí další nedostatek, kterým je fakt, že element MapGroup nepřijímá událost kliknutí myši, a zároveň zabraňuje do něj vnořeným elementům, aby ji přijaly. Mapový element tedy může zaznamenávat kliknutí, pouze pokud je deklarován osamoceně a není součástí většího celku elementů. Tento nedostatek řeší vytvoření dotazu nad prostorovou databází, kdy se zjišťuje zda souřadnice, na něž bylo na mapě kliknuto, nejsou uvnitř nebo v blízkosti nějakého geografického objektu, který se v databázi vyskytuje. V případě že ano, je navrácen jeho identifikátor a ten je následně vyhledán a objekt je označen pro manipulaci. Toto řešení se ukázalo jako funkční, ale je otázkou, jak je efektivní v případě, že se v databázi vyskytuje větší množství dat. Stejně řešení však nebylo možné realizovat na mobilním klientovi, kde sice není potřeba označovat objekty, ale je nutné vybírat popisy objektů pro zobrazení doplňujících informací. Z tohoto důvodu je tedy nutné označovat právě jména těchto popisů a nikoliv grafické objekty, které jim náležejí.

Původním záměrem při využití SVG jako formátu pro reprezentaci grafických dat bylo využít takto definovaná data rovnou jako obrázky a takto je i zobrazovat na klientské mapě, čímž by bylo dosaženo větší efektivity. Obrázky, a to i ve formátu SVG sice lze na mapě pomocí API zobrazovat, ty však mají stále stejnou velikost při všech úrovních přiblížení. To by bylo možné řešit škálováním obrázku podle aktuálního přiblížení mapy. Tuto vlastnost ale

API nenabízí. Proto je nutné SVG po přijetí parsovat a data interně reprezentovat stejným způsobem, jako je tomu u koordinátora. To na druhou stranu zaručuje přesně stejný tvar všech objektů u obou stran.

Ačkoli se využití XMPP protokolu ukázalo být vhodnou volbou, vyvstala z něj řada problémů, které souvisejí s knihovnou QXmpp³, což je jediná knihovna pro Qt, která s protokolem v době implementace umožňovala pracovat na všech třech požadovaných mobilních operačních systémech. Základním nedostatkem této knihovny je stav jejího rozpracování, který zdaleka není finální a knihovna tak nepodporuje některé očekávané funkce. To ve spojení s malou zdokumentovaností a malým množstvím příspěvků v diskuzích vedlo k některým problémům. Prvním je chybějící podpora pro rozšíření XEP-0080, které je jednou z klíčových složek aplikace. Třídy odvozené od této knihovny standardně neposkytují ani nezpracovaná data, která ale podle logovacího souboru byla obdržena. Řešením je vytvoření rozšíření, jejichž tvorba je standardně v rámci knihovny podporována, a v rámci tohoto rozšíření data získávat a zpracovávat. Dokumentace k tomu, jak toto rozšíření vytvořit, a že je možné jej vůbec využít však prakticky chybí.

Další problém vyvstal v souvislosti s překladem knihovny QXmpp na operační systém Symbian. Kompilace knihoven pro tento operační systém je obecně problematická, zejména proto, že kompilátor neposkytuje dostatečné informace o překladu a při jeho selhání, tak uživatel neví, co chybu způsobilo. Na rozdíl od ostatních platforem, pro něž byl projekt vyvíjen, vyžaduje Symbian využití dynamických knihoven a odlišný způsob kompilace, což se ani po několikadenním úsilí nezdařilo. Proto byl vytvořen nový projekt s klientskou částí systému, do nějž byly staticky přidány soubory knihovny QXmpp a celý projekt byl takto přeložen. Jedná se o problém, který na internetových diskuzních fórech již existuje, pravděpodobně ale nebyl prozatím vyřešen.

Posledním řešeným nedostatkem je spuštění klientské aplikace pro operační systém MeeGo. To sice funguje bez problému pokud je telefon připojen USB kabelem a aplikace je spuštěna z vývojového prostředí, jakmile je však vypnuta a kabel odpojen, není již dále možné ji zapnout s využitím vytvořené ikony. Je však možné ji spustit přes vestavěnou konzoli. V takovém případě ale pak nelze ukládat a zobrazovat značky a nelze tudíž vykreslovat objekty na mapě. Je pravděpodobné, že spuštěním aplikace se správnými parametry by fungovala stejně, jako po zapnutí z vývojového prostředí. Tyto parametry se nicméně nalézt nepodařilo.

³<http://qxmpp.googlecode.com/svn/doc/HEAD/html/index.html>

Kapitola 7

Testování a další vývoj

Tato kapitola se zabývá v první části testováním implementovaného softwaru z hlediska nároků na baterii a datové přenosy. Jsou zde popsány testy, které byly provedeny a analyzovány jejich výsledky. Na základě testů je pak navržena optimalizace systému. V druhé části se kapitola zabývá možnostmi, jak vytvořený program dále rozšířit, případně jak jej upravit tak, aby mohl sloužit pro jiné účely, než pro jaké byl vytvořen.

7.1 Výdrž baterie a datové nároky

Efektivita implementovaného systému je závislá na využití mobilních klientů. To může být ovlivněno omezenou kapacitou baterie mobilního telefonu a také kvalitou internetového připojení. Z toho důvodu byla provedena sada testů, které měly odhalit, jaká je výdrž jednotlivých zařízení při simulovaném provozu a jaké jsou datové nároky na internetové připojení. Kromě toho byl také vyhotoven test, jehož účelem je rozhodnout, jaký interval je vhodné zvolit pro získávání vlastní geografické polohy a její distribuci vzhledem k výdrži baterie.

První dva testy, celková výdrž baterií telefonů při provozu a nároky na připojení, byly realizovány na všech platformách. Maemo je reprezentováno telefonem Nokia N900, MeeGo telefonem Nokia N950 a Symbian telefonem Nokia X7. Provoz byl simulován vytvořením několika časovačů na straně koordinátora i mobilního klienta. Koordinátor každou minutu odesílal na klienty nový mapový objekt, kterým byla ve všech případech trasa s šesti uzly. Každé 4 minuty byla také odeslána na každého z klientů zpráva o délce 168 znaků. U klientů byl nastaven interval publikace vlastní polohy na 30 sekund a pohyb uživatele byl simulován programovou změnou pozice každou minutu, což s sebou přineslo i nutnost překreslení mapy. žádný z telefonů nebyl připojený k síti GSM. Tabulka 7.1 ukazuje srovnání výdrže všech telefonů spolu s velikostí akumulátoru každého z nich.

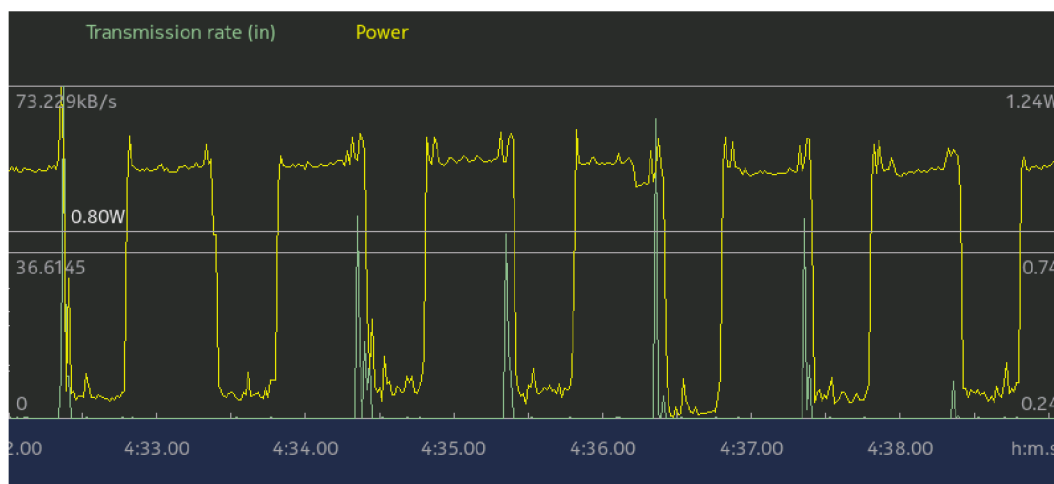
Z tabulky vyplývá, že u nejnovějšího zařízení v testu se systémem MeeGo se přibližný čas provozu aplikace na jedno nabití pohybuje kolem sedmi hodin. To se blíží době jedné směny

	Kapacita baterie	Doba provozu
Nokia N900	1320mAh	3h 38min
Nokia N950	1320mAh	7h 9min
Nokia X7	1200mAh	4h 53min

Tabulka 7.1: Doba výdrže jednotlivých telefonů při provozu

a tento čas je tedy prakticky dostačující. V případě, že by systém používala motorizovaná hlídka, nebylo by vybíjení baterie kritické, protože by mohla být nabíjena v automobilu. U pěší hlídky je výdrž telefonu podstatnější, a pro dosažení horizontu alespoň osmi hodin provozu je nutné provést optimalizace získávání a distribuce vlastních geografických souřadnic, čemuž se tato podkapitola věnuje dále. V případě zařízení s operačním systémem Symbian je nutné baterii pro dosažení osmihodinového provozu průběžně dobíjet. U zařízení se systémem Maemo je doba provozu poměrně nízká a zařízení se tak hodí spíše pro motorizované jednotky. Alternativou může být použití záložní baterie, která je v telefonu N900 vyměnitelná.

Druhý test probíhal simultánně s prvním a zabýval se nároky aplikace na datové připojení telefonu. Obrázek 7.1 ukazuje závislost požadavků na data a spotřeby energie na periodické změně polohy každou minutu. Data byla získána s pomocí aplikace Energy Profiler¹. Jak je možné vidět, vyvolá změna polohy nejprve zvýšenou spotřebu energie při výpočtu nové pozice mapy a následně jsou staženy nové mapové podklady. Z měření vyplývá že během první hodiny provozu aplikace stáhla 472,59 kB dat a odeslala 225,53 kB. Z toho 202,34 kB bylo staženo během spouštění aplikace. Další nárůst stažených dat byl již lineární s přírůstkem průměrně 282,41 kB. Po 4 hodinách provozu tak aplikace celkem stáhla a odeslala 2,12 MB dat. To je v současné době ve městech, kde má být systém nasazen primárně, splnitelný požadavek jak vzhledem k rychlosti připojení, tak vzhledem k omezení množství stažených a odeslaných dat.



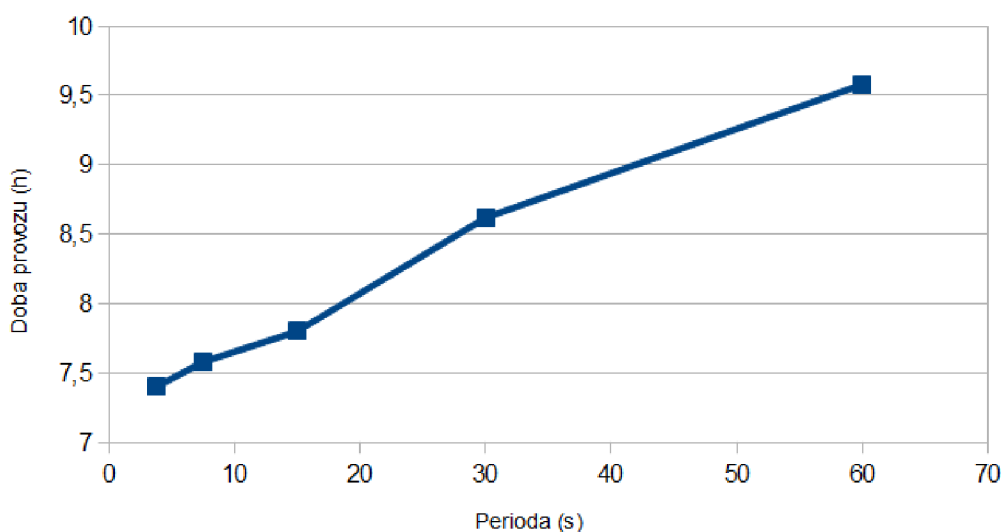
Obrázek 7.1: Závislost spotřeby energie na změně polohy v čase

Poslední test se zaměřil na vliv periody pro získání a distribuci geografické polohy klienta na celkový odhad spotřeby. Tento odhad byl získán pomocí aplikace Battery Usage² na telefonu Nokia N950. Na obrázku 7.2 je možné vidět, jak se pro jednotlivé časy mění odhad doby spotřeby. Z rozdílu mezi periodou 30 a 60 sekund, který dosahuje 58 minut provozu navíc je vidět, že získávání vlastní geografické polohy má značný vliv na výkon systému. Při časté změně je navíc nutné také častěji odesílat na server nová data, což se promítá taktéž v síťovém provozu. Na druhou stranu může mít příliš nízká doba změny polohy značný vliv na přesnost koordinace a následné možnosti analýzy pohybu jednotky.

U pěší jednotky se doba obnovy pozice 60 sekund při průměrné rychlosti pohybu 5

¹<http://store.ovi.com/content/73969>

²<http://store.ovi.com/content/138659>



Obrázek 7.2: Závislost výdrže baterie telefonu na periodě získávání pozice

km/h rovná přibližně 83 metrům, což je již velká odchylka. U motorizované jednotky je situace obdobná i pro desetkrát nižší periody. Z toho důvodu je pro nasazení systému klíčové testování při skutečném provozu, kdy vyjde najevo, jaký je správný poměr mezi výdrží mobilního zařízení, nároky na mobilní připojení a periodou pro získání a distribuci geografické polohy jednotek. Na základě testování bylo určeno, že vhodnou periodou pro pěší jednotku je čas mezi patnácti a třiceti sekundami a pro jednotku motorizovanou čas pod 7 sekund. Vzhledem k možnému nasazení systému pro pěší jednotky byla zvolena hodnota 20 sekund. Pro zvýšení efektivity je možné implementovat funkci pro výpočet optimálního intervalu podle relativní změny polohy během několika posledních měření.

7.2 Rozšíření a možná nasazení aplikace

V předchozí kapitole byla popsána funkcionalita, která byla v rámci této práce implementována. Vzhledem k zaměření systému a jeho možným aplikacím však existuje velké množství oblastí, v kterých je možné jej dále rozšiřovat a zdokonalovat. Tato kapitola se věnuje tomu, v jakých směrech je možné implementovaný software dále vyvíjet.

Oblasti, v nichž je možné program rozšiřovat, lze rozdělit na 3 části. První z nich je další přizpůsobování požadavkům a potřebám Policie ČR. Vývoj byl od počáteční specifikace značně ovlivněn konzultací s jejími zástupci a implementovaný software z velké části odpovídá tomu, jaké vlastnosti má koordinační systém pro danou oblast využití splňovat. Přesto je v případě, že bude systém někdy nasazen do provozu, nutné jej dále rozšířit. Požadavky, které jsou v současné době známy, ale které již nebyly do programu zahrnuty, vycházejí zejména klasifikace jednotek v terénu. Ty jsou v současné době všechny rovnocenné a jedinou jejich odlišností je zařazení k různým akcím.

V tomto smyslu je možné do systému zahrnout tzv. kódy typických činností, což je množina kódů, které se v praxi používají pro komunikaci s dispečerem pro označení stavu, v jakém se jednotka nachází. Možnostmi jsou například: "Jedu na místo", "Jsem na místě", "Věc vyřešena", nebo "Mám plánovanou přestávku". Tyto stavy jsou interně reprezentovány

vány kódem, který je možné poslat od jednotky ke koordinátorovi, a ten tak například nemůže přiřadit jednotku, která má přestávku, k akci dříve, než se tato jednotka opětovně ozve jako volná. V souvislosti s analýzou činnosti jednotek, zmíněných v předcházející kapitole, je navíc možné tyto informace použít k rozšíření znalostí o aktivitě jednotky v rámci jedné směny.

Další možností rozlišení různých jednotek je jejich zařazení k určitému obvodnímu oddělení, rozlišení na jednotky věnující se pouze dopravním nehodám a podobně. Ke každé jednotce lze kromě toho přiřadit výbavu, kterou má momentálně u sebe. Do ní může patřit kupříkladu hřebíkový pás, brokovnice, alkohol tester, defibrilátor, nebo pes. Tyto informace by mohla jednotka v rámci své klientské aplikace měnit a nastavovat a následně posílat ke koordinátorovi. Na základě toho by pak bylo možné na straně koordinátora jednotky filtrovat a zobrazovat pouze ty, které se nejlépe hodí k právě požadovanému úkolu.

Druhou oblastí možného rozšíření systému je implementace prvků, které nejsou přímo závislé na zvoleném zaměření programu. Těmito rozšířeními může být například vytvoření webového klienta pro koordinátora, který může nahradit stávající desktopovou verzi. Vytvoření webového klienta s sebou přináší výhodu zejména v souvislosti se snadnou přenositelností a kompatibilitou.

Jiným rozšířením je úprava části koordinátora tak, aby podporovala více simultánně operujících koordinátorů. Tato možnost byla zohledněna během návrhu, nicméně pro její efektivní realizaci je nutné přidat do systému funkce pro aktualizaci uživatelských rozhraní mezi jednotlivými koordinátory. V současné době je jedinou plně funkční variantou paralelní činnosti více koordinátorů rozdělení jednotek na dvě zcela oddělené části a přiřazení každé z částí jednomu z nich. Implementací distribuce změn mezi koordinátory a dílčími úpravami v systému je ale možné zmíněnou vlastnost do programu zavést.

Poslední oblastí, v níž je možné software rozšiřovat, je jeho přizpůsobení pro jiné zaměření. Jak lze vidět v kapitole 2, existuje v současné době řada podobných řešení a lze si velmi snadno představit různé další aplikace. Kupříkladu systém po úpravě mohou používat společnosti věnující se logistice, jako je DHL, nebo PPL. Ty jej mohou využít pro koordinaci svých kurýrů při rozvážení balíků ve městech.

Stejně využití lze nalézt i u dalších složek Integrovaného záchranného systému, jako je hasičská, nebo záchranná služba. Speciálním případem může být nasazení systému u horské záchranné služby, kdy by našla využití vlastnost, zmíněná v kapitole 6, díky které je možné získávat geografická data o uživateli, aniž tito uživatelé potřebují vytvořenou klientskou aplikaci. V takovém případě si může turista stáhnout volně dostupnou aplikaci Azimuth do svého mobilního telefonu a u záchranné služby potvrdit své začlenění do systému. Koordinátor záchranných služeb následně uvidí jeho geografickou polohu na mapě a v případě, že například dostane ohlášen pád laviny, může ihned zkontrolovat, zda se v zasažené oblasti turista nacházel, a v případě že ano, ihned začne koordinovat záchranné jednotky, které využívají plnohodnotného klienta.

Jistě ne poslední možností, jak systém využívat v praxi, je jeho rozšíření jednotlivcům a domácnostem, kdy může sloužit ke kontrole pohybu dětí, nebo hlídání vlastního majetku, jako je automobil, nebo samotný mobilní telefon. Možností pro využití vyvinutého softwaru je velké množství. Některé aplikace nevyžadují příliš velké úpravy ve stávajícím kódu, pro jiné je nutné systém přizpůsobit.

Kapitola 8

Závěr

Cílem diplomové práce bylo v první části prostudovat problematiku tvorby aplikací pro operační systém Maemo se zaměřením na použití map a vzájemnou komunikaci zařízení. Tomu se věnuje jednak kapitola 2, v níž jsou představeny aplikace, jejichž zaměřením je právě komunikace zařízení spolu s využíváním map. Problematiku vývoje mobilních aplikací pro Maemo shrnuje kapitola 3, která kromě toho představuje i další dvě platformy, Symbian a MeeGo, pro které je výsledný systém taktéž implementován.

Další částí diplomové práce bylo navrhnout vhodnou aplikaci, která využívá mapy a komunikaci s dalšími zařízeními a experimentovat s použitelnými technologiemi. Tomu jsou věnovány kapitoly 4 a 5. V první jmenované jsou představeny technologie, které lze v aplikaci použít. Na základě jejich vlastností a požadavků, které na ně klade aplikace, jsou vybrány ty, které se pro implementaci hodí nejlépe. Ve stejné kapitole jsou taktéž představeny experimenty s komponentami pro zobrazování map a různými mapovými podklady. Druhá kapitola je zaměřena přímo na samotný návrh se zaměřením na případy použití a architekturu celého systému.

Po výběru technologií a návrhu programu bylo cílem práce tento návrh implementovat a následně demonstrovat použitelnost vytvořeného řešení. Tomu je věnována kapitola 6, kde je představena implementovaná funkcionalita, uživatelské rozhraní a jsou zmíněny problémy, které bylo nutné řešit. Součástí popisu funkcionality je i výčet několika modelových situací, na nichž jsou implementované funkce představeny.

Poslední částí při vypracování této práce bylo zhodnocení dosažených výsledků a zvážení možnosti, jak v projektu dále pokračovat. Těmto cílům je věnována kapitola 7, kde jsou v rámci zhodnocení výsledků provedeny testy při simulovaném provozu systému za účelem analýzy nároků aplikace na baterii mobilních zařízení a změření požadavků na datové přenosy. Následně jsou popsány 3 možné směry dalšího vývoje projektu včetně jeho možného nasazení pro jiné účely, než pro které byl původně vytvářen. V rámci práce bylo také vytvořeno prezentační video, které ukazuje funkcionalitu systému v provozu.

Cíl práce, vytvořit víceuživatelskou mapovou aplikaci pro mobilní zařízení, byl splněn. Aplikace kromě vzájemné komunikace spolupracují také se statickým klientem v podobě koordinátora. Pro aplikaci byl zvolen implementační případ, jímž je koordinační systém pro Policii České republiky. Z konzultací s jejími zástupci vyplynulo, že jsou z hlediska funkcionality a použitelnosti s aplikací spokojeni, a tak bude jedním s cílů této práce do budoucna nasazení výsledného systému do reálného provozu. Práce se dne 26.4.2012 zúčastnila soutěže EEICT, kde se umístila na prvním místě v kategorii Informační systémy.

Literatura

- [1] Aron, L.: *Plánovač polohy založený na poloze pro N900*. Bakalářská práce, Brno, FIT VUT v Brně, 2011, 2011.
- [2] Eliáš, B. F.: *Vylepšení podpory REST v JBossESB*. Diplomová práce, Brno, FIT VUT v Brně, 2011, 2011.
- [3] Environmental Systems Research Institute, Inc: *RFC 6120. The Secure HyperText Transfer Protocol*. 2011., Dostupné z WWW: <<http://xmpp.org/rfcs/rfc6120.html>>, [Online; cit. 2011-12-21].
- [4] Hildebrand, J.; Saint-Andre, P.: XEP-0080: User Location. <<http://xmpp.org/extensions/xep-0080.html>>, [Online; cit. 2011-12-25].
- [5] IETF Trust: *ESRI Shapefile Technical Description*. 1998, Dostupné z WWW: <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>>, [Online; cit. 2011-12-21].
- [6] The Internet Society: *RFC 2616. Hypertext Transfer Protocol - HTTP/1.1*. 1999., Dostupné z WWW: <<http://www.w3.org/Protocols/rfc2616/rfc2616.html>>, [Online; cit. 2011-12-22].
- [7] The Internet Society: *RFC 2660. The Secure HyperText Transfer Protocol*. 1999., Dostupné z WWW: <<http://rfc.askapache.com/rfc2660/rfc2660.txt>>, [Online; cit. 2011-12-22].
- [8] The Internet Society: *RFC 2965. HTTP State Management Mechanism*. 2000., Dostupné z WWW: <<http://tools.ietf.org/html/rfc2965>>, [Online; cit. 2011-12-22].
- [9] Jonáš, J.: *Aplikace pro cyklisty v prostředí Flash (Air)*. Bakalářská práce, Brno, FIT VUT v Brně, 2008, 2008.
- [10] Kubáni, T.: *Zabezpečenie webových služieb*. Bakalářská práce, Brno, FEKT VUT v Brně, 2011, 2011.
- [11] Kulička, B. V.: *VoIP in jabber client*. Diplomová práce, Brno, FIT VUT v Brně, 2011, 2011.
- [12] Larman, C.: *Applying UML and patterns : introduction to object-oriented analysis and design and interactive development*. Prentice-Hall, 2005, iSBN 978-0-13-148906-2.
- [13] Molkenin, D.: *The Book of Qt4*. Open Source Press, 2007, iSBN 978-3-937514-12-3.

- [14] Open Geospatial Consortium: *Geography Markup Language*. 2010, Dostupné z WWW: <<http://www.opengeospatial.org/standards/gml>>, [Online; cit. 2011-12-23].
- [15] Pouliček, B. Z.: *Víceuživatelská mapová aplikace pro kapesní zařízení*. Diplomová práce, Brno, FIT VUT v Brně, 2010, 2010.
- [16] Richardson, L.; Ruby, S.: *RESTful Web Services*. O'Reilly Media, Inc., 2007, iISBN 978-0-596-52926-0.
- [17] Rischpater, R.; Zucker, D.: *Beginning Nokia Apps Development*. Apress, 2010, iISBN 978-1-4302-3177-6.
- [18] Saint-Andre, P.; Smith, K.; Tronçon, R.: *XMPP: The definitive guide*. O'Reilly Media, Inc., 2009, iISBN 978-0-596-52126-4.
- [19] Simhachalam, M.: Securing REST Web Services With OAuth. <<http://developers.sun.com/identity/reference/techart/restwebservices.html>>, [Online; cit. 2011-12-28].
- [20] Tuček, J.: *Geografické informační systémy*. Computer Press, 1998, iISBN 978-80-7226-091-0.
- [21] World Wide Web Consortium: *Simple Object Access Protocol (SOAP) 1.1*. 2000, Dostupné z WWW: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>, [Online; cit. 2012-05-04].
- [22] World Wide Web Consortium: *SOAP Version 1.2 Part 1: Messaging Framework*. 2007, Dostupné z WWW: <<http://www.w3.org/TR/soap12-part1/>>, [Online; cit. 2012-05-04].
- [23] World Wide Web Consortium: *SOAP Version 1.2 Part 2: Adjuncts*. 2007, Dostupné z WWW: <<http://www.w3.org/TR/soap12-part2/>>, [Online; cit. 2012-05-04].
- [24] WWW stránky: Capabilities (Symbian Signed). <http://www.developer.nokia.com/Community/Wiki/Capabilities_%28Symbian_Signed%29>, [Online; cit. 2011-12-29].
- [25] WWW stránky: Categorical Index Of SQLite Documents. <<http://www.sqlite.org/docs.html>>, [Online; cit. 2012-05-03].
- [26] WWW stránky: Extensible Messaging and Presence Protocol (XMPP): Core. <<http://xmpp.org/rfcs/rfc3920.html>>, [Online; cit. 2012-05-03].
- [27] WWW stránky: Facebook Chat API. <<https://developers.facebook.com/docs/chat/>>, [Online; cit. 2012-05-01].
- [28] WWW stránky: Installing and Signing. <http://library.developer.nokia.com/index.jsp?topic=/S60_5th_Edition_Cpp_Developers_Library/GUID-35228542-8C95-4849-A73F-2B4F082F0C44/sdk/doc_source/guide/cpp-intro/HandsOnCpp.html>, [Online; cit. 2011-12-29].

- [29] WWW stránky: Maemo Development. <<http://maemo.org/development/>>, [Online; cit. 2012-01-04].
- [30] WWW stránky: Maemo SDK. <http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Development_Environment/Maemo_SDK>, [Online; cit. 2011-12-22].
- [31] WWW stránky: Maemo SDK+ User Guide. <<http://maemo-sdk.garage.maemo.org/user-guide.html>>, [Online; cit. 2011-12-22].
- [32] WWW stránky: MeeGo Wiki. <<http://wiki.meego.com/Glossary>>, [Online; cit. 2011-12-28].
- [33] WWW stránky: MySQL 5.5 Reference Manual. <<http://dev.mysql.com/doc/refman/5.5/en/index.html>>, [Online; cit. 2012-01-03].
- [34] WWW stránky: Nokia Smart Installer for Symbian. <http://www.developer.nokia.com/Community/Wiki/Nokia_Smart_Installer_for_Symbian>, [Online; cit. 2012-01-05].
- [35] WWW stránky: OpenStreetMap. <<http://www.openstreetmap.org/>>, [Online; cit. 2011-12-24].
- [36] WWW stránky: PostGIS 1.5.3 Manual. <<http://postgis.refractions.net/documentation/manual-1.5/>>, [Online; cit. 2011-12-28].
- [37] WWW stránky: PostgreSQL 9.1.2 Documentation. <<http://www.postgresql.org/docs/9.1/interactive/index.html>>, [Online; cit. 2011-12-28].
- [38] WWW stránky: Qt Mobility Project Reference Documentation. <<http://doc.qt.nokia.com/qtmobility-1.2>>, [Online; cit. 2012-01-07].
- [39] WWW stránky: Scalable Vector Graphics (SVG) 1.1 (Second Edition). <<http://www.w3.org/TR/SVG/>>, [Online; cit. 2012-01-02].
- [40] WWW stránky: Symbian Signed. <<https://www.symbiansigned.com/>>, [Online; cit. 2011-12-28].
- [41] Zander, T.: Symbian development using Linux. <<http://labs.qt.nokia.com/2010/04/21/symbian-development-using-linux/>>, [Online; cit. 2012-01-05].

Příloha A

Obsah DVD

/Coordinator/Diplomka/* - zdrojové kódy mobilní aplikace
/Coordinator/Diplomka-Central/* - zdrojové kódy aplikace koordinátora
/Coordinator/dbschema.sql - inicializační skript pro databázový systém PostgreSQL
/Coordinator/README.txt - manuál pro spuštění programu
/video/video.avi - prezentační video implementované aplikace
/thesis/* - zdrojové soubory této práce
/projekt.pdf - tato práce ve formátu pdf