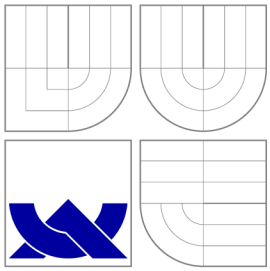
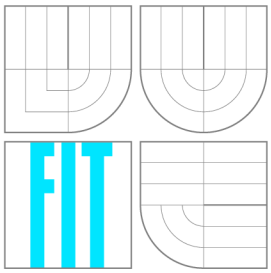


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

APLIKACE PRO ZABEZPEČENOU MOBILNÍ KOMUNIKACI

APPLICATION FOR SECURE MOBILE COMMUNICATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAKUB JEŘÁBEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. PAVEL OČENÁŠEK, Ph.D.
SUPERVISOR

BRNO 2013

Abstrakt

Cílem této bakalářské práce je navrhnout a vytvořit mobilní aplikaci, která bude umožňovat zasílání textových zpráv s důrazem na jejich důvěrnost. Aplikace má umožňovat správu kontaktů, jejich zařazování i do více skupin, zasílání hromadných zpráv. Dále má aplikace umožňovat jejím uživatelům sdílet svou pozici pomocí GPS modulu. Celé řešení využívá architekturu klient-server.

Ve své práci provádím analýzu stávajících řešení, na jejímž základě navrhuji specifikaci výsledné aplikace, jejíž implementace je v práci také popsána. Výsledkem mé činnosti je funkční aplikace pro operační systém Android splňující zadání práce.

Abstract

The aim of this bachelor's thesis is to design and implement a mobile application for instant messaging with the accent on confidentiality. The application will provide the management of contacts, the management of contact groups and sending group messages. What is more, the application should allow users to share their position via the device's GPS module. In this thesis, the client-server architecture is used.

The thesis contains an analysis of existing solutions, based on which I propose a specification of the resulting application. Its implementation is also described. The result of my activity is a working application for the mobile operation system Android which complies with the assignment of the thesis.

Klíčová slova

zasílání zpráv, šifrování, zabezpečení, klient–server, SSL, Android, Java, MySQL, SQLite

Keywords

instant messaging, cyphering, security, client–server, SSL, Android, Java, MySQL, SQLite

Citace

Jakub Jeřábek: Aplikace pro zabezpečenou mobilní komunikaci, bakalářská práce, Brno, FIT VUT v Brně, 2013

Aplikace pro zabezpečenou mobilní komunikaci

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavla Očenáška, Ph.D. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Jeřábek
12. května 2013

Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Pavlu Očenáškovi, Ph.D. za přínosné a cenné rady, které mi poskytoval v průběhu řešení mé práce a panu Ladislavu Kudláčkovi za poskytnutí serveru pro testování aplikace.

Obsah

Kapitola 1

Úvod

V posledních několika letech můžeme sledovat téměř drtivý nástup tzv. chytrých telefonů, které začínají odsunovat klasické mobilní telefony do pozadí. Svědčí o tom čísla o prodejích od tiskových mluvčí českých operátorů. Na konci roku 2012 připadlo na chytré telefony 71 % ze všech prodaných zařízení u společnosti Telefonica O_2, u operátora T-Mobile tvořil podíl chytrých telefonů na prodejích 80 % na začátku roku 2013 [1].

Ve Spojených státech amerických podíl chytrých telefonů mezi uživateli stabilně roste a v lednu roku 2013 dosahoval 55 % [2].

Jaké možnosti poskytují dnešní chytré telefony svým uživatelům? Přístup do e-mailové schránky, internetového bankovnictví, na sociální sítě, do obchodu s mobilními aplikacemi či hudbou a také možnost komunikovat prostřednictvím chatovacích aplikací. Se stále se zlepšujícími se hardwarovými parametry se z chytrých telefonů stávají také fotoaparáty a videokamery.

Při krádeži či ztrátě telefonu tak má zloděj či nálezce přístup téměř k celému soukromí majitele, příp. i jeho firemním záležitostem. Snad s výjimkou aplikace pro přístup do internetového bankovnictví žádná z výše uvedených nepožaduje autentizaci uživatele a ztráta mobilního telefonu se tak může téměř rovnat ztrátě notebooku.

Dalším způsobem, jakým lze narušit cizí soukromí je útokem prostřednictvím sítě Internet. Nejsnadnější je využít aplikací speciálně vytvořených pro útok na konkrétní aplikaci, např. Facebook [3] či WhatsApp Messenger [16], kde k získání kontroly nad účtem postačí řádově sekundy a několik kliknutí.

S rozmachem internetového připojení v telefonech také vzrůstá počet zaslaných zpráv skrze aplikace na úkor SMS zpráv. Podle zveřejněné studie bylo v roce 2012 odesláno téměř 19 miliard zpráv denně oproti 17,6 miliardy SMS zprávám [4]. Položme si otázku, kolik z těchto zpráv bylo nezabezpečených? Kolik z těchto zpráv se jakýmkoliv způsobem dostalo do cizích rukou?

Při analýze stávajících řešení, která je popsána v kapitole 2, jsem nenašel žádnou aplikaci pro textovou komunikaci, která by kladla důraz na zabezpečení zpráv, a to nejen při jejich přenosu, ale také v zařízení. Výjimku tvoří plugin pro zařízení BlackBerry, který je popsán v 2.2.1 a stal se inspirací pro tuto práci. Třetí kapitola je věnována specifikování požadavků, které by měla aplikace mít, aby dobře sloužila svému účelu, tedy zabezpečení zpráv. Další dvě kapitoly jsou věnovány návrhu aplikace a její implementaci. V kapitole 6 diskutuji dosažené výsledky a další možný rozvoj aplikace.

Kapitola 2

Analýza existujících řešení

2.1 Aplikace pro OS Android

2.1.1 Cloak SMS

Aplikace Cloak SMS Free[1] nabízí zasílání a příjem zpráv zašifrovaných šifrou AES[2]. Šifrovací klíč je nutné si předávat s druhou stranou jiným komunikačním kanálem mimo tuto aplikace, tedy např. ústní dohodou, telefonickým hovorem nebo SMS zprávou. Samotné šifrované zprávy jsou odesílány pomocí SMS zpráv a při příjmu jsou aplikací odchyceny a nejsou tak doručeny mezi ostatní nešifrované SMS zprávy v telefonu. Placená verze aplikace umí stejnou šifrou zašifrovat také soubory. [5]

2.1.2 Crypt Haze

Crypt Haze[3] taktéž využívá šifru AES s 256-bitovým klíčem. Aplikace je podobná výše uvedené Cloak SMS, zasílat zašifrované zprávy lze nejen pomocí SMS zpráv, ale i e-mailu. Rozpoznání příchozí zašifrované zprávy však funguje pouze u SMS. [6]

2.2 Aplikace pro OS BlackBerry

2.2.1 Ekboo Crypto Chat

Ekboo Crypto Chat[4] je plugin pro aplikaci BlackBerry Messenger (BBM). BlackBerry Messenger je proprietární aplikace pro zasílání krátkých zpráv umožňující komunikaci mezi dvěma zařízeními s OS BlackBerry využívající pro zasílání zpráv servery společnosti Research in Motion (RIM). Standardně nejsou zasílaná data nijak šifrována.

Plugin rozšiřuje BBM o klíčové vlastnosti. Mezi stěžejní patří, vyjma samotného šifrování zpráv:

- možnost nouzového nenápadného vymazání všech konverzací,
- zaslání zprávy s omezenou časovou platností,

[1] URL <https://play.google.com/store/apps/details?id=sms.encryptor.v2.free>

[2] Advanced Encryption Standard – symetrická bloková šifra, používaná americkými federálními úřady pro šifrování nejtajnějších dokumentů [9]

[3] URL <https://play.google.com/store/apps/details?id=net.rehacktive.cryptdroid>

[4] URL <http://appworld.blackberry.com/webstore/content/62871/>

- přístup do aplikace chráněn PIN kódem,
- šifrovaný přenos souborů 256-bitovou symetrickou šifrou AES.

K šifrování zpráv používá plugin Ekboo asymetrickou šifru RSA s klíči o délce 512 bitů. Výměna veřejných klíčů je možná i za běhu, což zvyšuje zabezpečení. Způsob výměny klíčů není veřejně znám. Mezi další vlastnosti pluginu patří vymazání konverzací v příjemcově zařízení, zavření aktuálních diskuzí při neaktivitě a notifikace při doručení i přečtení zprávy.

Plugin neukládá žádná data do telefonu a pokud dojde k restartu aplikace, veškerá data jsou ztracena.

Jednou z posledních novinek je možnost fyzické výměny ověřovacího klíče pomocí QR kódu pro prevenci Man-in-the-middle útoku. Z toho usuzuji, že aktuálně není komunikace proti tomuto útoku chráněna. [7]

2.3 Aplikace pro OS iOS

2.3.1 ChatSecure

ChatSecure[5] je otevřená aplikace zabezpečující komunikaci přes protokoly XMPP (Google Talk, Jabber) nebo Oscar (AIM) pomocí protokolu Off-the-Record. Tomuto protokolu se více věnuji v sekci 2.5. [8]

2.4 Aplikace pro OS Windows Phone

2.4.1 4UrEyezOnly

Autoři aplikace 4UrEyezOnly[6] nezveřejnili konkrétní způsob šifrování dat, prohlašují však o ní, že se jedná o šifrování na armádní úrovni používané federálními úřady. Z toho se dá usoudit, že se s největší pravděpodobností jedná o symetrickou šifru AES.

Přidanou hodnotou aplikace je použití Digital Rights Management (DRM) ochrany zprávy, což znemožňuje její kopírování, přeposílání a ukládání mimo aplikaci. Dalším bonusem oproti jiným aplikacím je možnost kdykoliv vymazat odeslanou zprávu ze zařízení příjemce, dokonce i po jejím přečtení. [10]

2.5 Multiplatformní řešení

2.5.1 WIZ Messenger

Tato aplikace pro zasílání zpráv dostupná pro OS Android, iOS a Windows Phone šifruje veškeré zprávy pomocí protokolu HTTPS, tedy samotná data jsou šifrována za využití protokolu SSL[7] nebo TLS[8]. Wiz Messenger[9] nabízí skupinové konverzace s možností přidávat další účastníky za běhu, sdílení fotografií a push notifikace. Aplikace umožňuje připravit zprávu ve stavu bez připojení k síti, tato bude automaticky odeslána po připojení k síti. [11]

[5] URL <https://itunes.apple.com/us/app/chatsecure-encrypted-secure/id464200063?mt=8>

[6] URL <http://www.windowsphone.com/cs-cz/store/app/4ureyezonly/21c523c7-7869-e011-81d2-78e7d1fa76f8>

[7] Secure Sockets Layer, definováno v RFC 6101, URL <http://tools.ietf.org/html/rfc6101>

[8] Transport Layer Security, definováno v RFC 5246, URL <http://tools.ietf.org/html/rfc5246>

[9] URL <https://play.google.com/store/apps/details?id=com.dynmark.wasp>

2.5.2 Kryptos

Aplikace Kryptos[10], dostupná pro OS Android, BlackBerry a iOS, neslouží pro zasílání šifrovaných zpráv, ale pro šifrování hovorů uskutečňovaných přes VoIP technologii. Pro šifrování hovoru používá symetrickou šifru AES s 256-bitovým klíčem, pro přenos klíče využívá asymetrickou šifru RSA s klíči dlouhými 2048 bitů.

Hovory jsou uskutečňovány pomocí peer-to-peer spojení, nikoliv přes server a lze je pochopitelně uskutečňovat pouze mezi uživateli aplikace.

Pro běh aplikace je nutná registrace, která obnáší celé jméno, Kryptos ID – unikátní uživatelské jméno, heslo a e-mailovou adresu. Po přihlášení je možné spravovat vlastní adresář, který sestává z Kryptos ID a jména. [12]

2.5.3 WhatsApp Messenger

WhatsApp Messenger[11] je jedna z nejstahovanějších aplikací [13] dostupná pro Android, BlackBerry, iOS, Symbian, Windows Phone a Nokia Series 40 a 60. Jedná se o klasického klienta pro zasílání zpráv, který umožňuje skupinovou komunikaci, sdílení libovolné GPS polohy a fotografií či videí.

Sdílení polohy je realizováno způsobem, kdy uživatel vybere na mapě libovolný bod a ten je poté odeslán druhé straně. Jako výchozí bod je zvolena aktuální poloha. Není tedy možné, aby si člověk zobrazil polohu druhé osoby. [14]

Od verze 2.8.3, která vyšla v srpnu roku 2012, aplikace šifruje veškerou komunikaci, bohužel však není známo, jakým způsobem. Do té doby aplikace zasílala konverzace v podobě otevřeného textu a bylo možné komunikaci jednoduše odposlouchávat. Pro usnadnění odposlouchávání vznikly speciální aplikace nazvané Whatsapp Sniffer [16]. [15]

2.5.4 Secret Message Elite

Tato aplikace je k dispozici pro OS Android, iOS a Windows Phone 7 a umožňuje šifrování jak textu, tak i fotografií. Zašifrovaná data je možné sdílet na sociální sítě Facebook, Twitter a Google+. Druhý způsob zaslání zprávy je více klasický: e-mailem či SMS zprávou. V případě použití Facebooku či e-mailu má příjemce usnadněnou situaci tím, že je ke zprávě připojen odkaz na server s předvyplněným zašifrovaným textem a stačí tedy zadat pouze správný dešifrovací klíč. V jiných případech je nucen příjemce kopírovat text a vkládat jej do aplikace, což nemusí být vždy komfortní.

Samotná aplikace neslouží ke komunikaci, ale pouze zašifrování, resp. dešifrování zprávy. Opět je zde použita symetrická bloková šifra AES se 128 či 256-bitovým klíčem.

Výhodou této aplikace je její tzv. Stealth Mode, který znamená, že aplikace není viditelná v seznamu aplikací a otevřít ji lze pouze vytočením čísla 3666. Další výhodu spatřuji v tom, že přidává do šifrovaných dat tzv. sůl, čímž zvyšuje odolnost vůči slovníkovým útokům nebo použití duhových tabulek[12] (rainbow tables). [17]

[10] URL <https://play.google.com/store/apps/details?id=com.appetizermobile.kryptos>

[11] URL <https://play.google.com/store/apps/details?id=com.whatsapp>

[12] Duhové tabulky obsahují předvypočtené hodnoty, které jsou určeny pro velmi rychlé prolamování hashovacích funkcí.

2.5.5 Off-the-Record Messaging

Off-the-Record Messaging[13], dále jen OTR, je bezpečnostní protokol, díky kterému lze dosáhnout silného šifrování diskuzí skrze různé komunikační protokoly. K tomuto používá OTR kombinaci symetrické šifry AES, hashovací funkce SHA-1 a Diffie-Hellmanovu výměnu klíčů.

Autoři v protokolu použili dvě méně obvyklé vlastnosti, první z nich je PFS (Perfect Forward Secrecy), která díky použití dočasných klíčů pro šifrování zajišťuje, že v případě jejich prolomení nedojde ke zkompromitování i předchozích relací. Druhou vlastností je „Deniable authentication“, díky které není možné jednoznačně ztotožnit konkrétního odesílatele či příjemce s konkrétní zprávou.

Nevýhodou OTR protokolu je nemožnost zasílat zprávy uživatelům, kteří nejsou právě připojeni. To je způsobeno nutností výměny šifrovacích klíčů před samotnou konverzací. Protokol OTR také prozatím nepodporuje skupinové konverzace. [18, 19]

2.6 Aplikace pro desktopové OS

2.6.1 ICQ

Desktopová aplikace ICQ[14] má ve svém nastavení možnost zvolit šifrování komunikace za využití protokolu SSL. Aplikace umožňuje kontakty třídit do skupin, přičemž zde platí pravidlo, že jeden kontakt může být pouze v jedné skupině.

Autoři aplikace vyvinuli také mobilní verzi své aplikace, ta v nastavení nemá možnost volby šifrování prostřednictvím protokolu SSL a je tedy otázkou, zda je tento protokol použit či nikoliv. Z dostupných zdrojů se mi nepodařilo tuto informaci získat. Mobilní aplikace umožňuje i sdílení GPS pozice, to je realizováno stejným způsobem jako u aplikace WhatsApp Messenger, jak jsem popsal v sekci 2.5.3.

2.6.2 Pluginy pro Miranda IM

Aplikace Miranda IM[15] je jedním z otevřených klientů, kteří podporují několik různých protokolů. Vzhledem k rozšířené komunitě existuje pro tohoto komunikačního klienta velká nabídka pluginů a část z nich je zaměřená na bezpečnost. Většina pluginů používá pro šifrování komunikace protokol OTR, nebo GnuPG. Protokol OTR je již popsán v části 2.5.5.

2.6.3 GNU Privacy Guard

GnuPG[16] je volně šiřitelná a patenty neomezená varianta k PGP[17]. Šifrování je založeno na páru asymetrických klíčů vygenerovaných uživatelem. K zaslané zprávě je možné připojit digitální podpis, který vznikne zašifrování otisku zprávy soukromým klíčem odesílatele. Příjemce vytvoří stejným způsobem vlastní otisk zprávy, dešifruje veřejným klíčem příjemce jeho otisk a tyto dva porovná. V případě shody je zaručena autenticita a integrita zprávy. [21]

[13] URL <http://www.cypherpunks.ca/otr/>
[14] URL <http://www.icq.com/>
[15] URL <http://www.miranda-im.org/>
[16] URL <http://www.gnupg.org/>
[17] Pretty Good Privacy, standardizováno pod názvem OpenPGP, URL <http://tools.ietf.org/html/rfc4880>

Kapitola 3

Specifikace požadavků na aplikaci

3.1 Zabezpečení spuštění aplikace

Kromě zabezpečení přenosu zpráv přes síť Internet je nutné zabezpečit aplikaci proti nežádoucímu vstupu jiných osob. Toho lze dosáhnout několika způsoby, příp. jejich kombinací.

Mezi tradiční jistě patří vyžadování zadání PIN kódu nebo hesla při spuštění aplikace. Jako ochranu proti útoku hrubou silou na toto opatření je vhodné zavést omezení na počet pokusů v určitém časovém úseku.

Aplikace Secret Message Elite, jak bylo popsáno v sekci 2.5.4, není volitelně viditelná v seznamu nainstalovaných aplikací a pro její otevření je nutné vytočit číslo 3666. Obdobně funguje i antivirus Avast pro OS Android, který jde ještě dále v zabezpečení a pro jeho otevření je nutné vytočit uživatelův PIN kód. I toto je tedy jeden ze způsobů, jakým by mohlo být zabezpečeno spuštění aplikace.

3.2 Zabezpečení zpráv při přenosu

Aplikace by měla nabízet takový šifrovací aparát, aby přenášená data byla v ideálním případě odolná vůči všem známým útokům. V dnešní době plné odposlechů je kladen velký důraz na utajení důležitých informací a tato aplikace by měla být jeden z možných prostředků, jak toho docílit.

3.3 Další bezpečnostní funkce aplikace

Při zasílání zprávy by měla být možnost přidat ke zprávě časovač, který způsobí její smazání z příjemcova zařízení. Varianty jsou dvě, buď může dojít k jejímu smazání po uplynutí navoleného času od přijetí, nebo od přečtení. Tímto se může odesílatel pojistit, že se zpráva později nedostane k někomu jinému, pokud si například není zcela jist spolehlivostí protějšku.

Pokud se uživatel aplikace dostane do situace, kdy bude nucen zadat svůj PIN kód, bude se mu hodit možnost nastavení falešného PIN kódu. Po jeho zadání aplikace vymaže veškerou komunikaci a normálně se spustí. V tomto momentě by bylo také vhodné nestahovat nové zprávy. Pro zvýšení důvěryhodnosti falešného PIN kódu by mohl uživatel mít možnost u zprávy nastavit příznak „nemazat v nouzi“. Obdobnou funkcionalitu můžeme vidět například u aplikace TrueCrypt[1], kde se nazývá hodnověrné popření, anglicky „plausible

[1] URL <http://www.truecrypt.org/>

deniability“. Pomocí této aplikace je možné vytvořit šifrovaný oddíl, který bude obsahovat dva další šifrované oddíly se svými vlastními přístupovými hesly. Po zadání jednoho z hesel není možné jakkoliv prokázat existenci dalšího oddílu [25].

Ke zvýšení zabezpečení zpráv přispěje také zvolená možnost neukládat příchozí či odchozí zprávy.

Pokud to nebude jedna z vlastností použitého zabezpečení, mělo by docházet k pravidelné obměně šifrovacích klíčů z důvodů zajištění PFS, tedy nemožnosti dešifrovat kompromitovaným klíčem předchozí konverzaci.

Upozorňování na příchozí zprávy by mělo být nastavitelné, a to především úroveň diskrétnosti. Pro některé uživatele by nemuselo být žádoucí, aby se jim ve stavové liště zobrazovala ikona nové zprávy doprovázená zvukem a vibracemi. Zde se nabízí možnost zobrazit ve stavové liště např. stav nabití baterie v procentech jako skryté upozornění na nepřečtené zprávy.

3.4 Ostatní funkce aplikace

3.4.1 Sledování polohy kontaktu

Uživatel aplikace by měl mít možnost automaticky sdílet se svým protějškem svou GPS pozici. Tato volba musí být volitelná a pozice by měla být přenášena šifrovaně. Možné využití spatřuji pro rodiče, kteří chtějí mít přehled, kde se momentálně nachází jejich dítě.

K odesílání polohy by mělo docházet automaticky v pravidelném časovém intervalu. Poslední poloha by měla být uložena na serveru spolu s časovým razítkem a měla by být zasílána ostatním uživatelům na vyžádání, nikoliv automaticky. V klientské aplikaci by měla být k dispozici pouze poslední známá poloha druhé osoby, bez historie.

3.4.2 Tvorba zprávy

Při psaní zprávy by měla být k dispozici co největší plocha displeje, využívat by se měla systémová klávesnici, na kterou je uživatel zvyklý z jiných aplikací. Délka zprávy by neměla být žádným způsobem omezena. Veškeré nastavení týkající se připravované zprávy by mělo být na jedné obrazovce spolu s tlačítkem na její odeslání, resp. zrušení.

3.4.3 Správa skupin

S tím, jak aplikace bude umožňovat zaslání hromadné zprávy, je žádoucí, aby bylo možné kontakty přiřazovat do vlastních skupin. Měl by zde být vztah 1:N, tedy jeden kontakt se může nacházet ve více skupinách. Vytvořeným skupinám by mělo být možné zaslat jednoduše hromadnou zprávu.

3.4.4 Načasování odeslání zprávy

Funkce načasování odeslání zprávy na určitý den a hodinu jistě přispěje k většímu uživatelskému komfortu. Nabízí se dvě možnosti, jak toto realizovat. První je připravení zprávy v klientské aplikaci a v uvedený čas její zaslání přes server příjemci. Druhý způsob je odeslání zprávy na server, kde bude tato uložena a odeslána příjemci až v nastavený datum a čas.

Druhé řešení má tu výhodu, že pro jeho realizaci není potřeba být připojen k síti v požadovaném čase odeslání.

3.5 Vzhled a chování aplikace

Aplikace by neměla nikterak svým grafickým uživatelským rozhraním a chováním vybočovat z řady podobných aplikací. To z důvodu, že uživatel bude jistě očekávat chování, na které je zvyklý z podobných aplikací. Ovládání aplikace by mělo být přímočaré a jednoduché. Zasílané i přijaté zprávy by se měly seskupovat do konverzací obdobným způsobem, jako je tomu běžné u textových zpráv na dnešních chytrých telefonech.

Šifrováním a dešifrováním zpráv by neměl být uživatel zatěžován, vše by se mělo dít na pozadí v minimálním možném čase (avšak nikoliv na úkor zabezpečení) a bez uživatelova vědomí.

Aplikace by také měla splňovat nepsané standardy chování pro OS Android. Kupříkladu po stisku hardwarového tlačítka zpět by mělo dojít k jejímu odeslání na pozadí a rozhodnutí o jejím ukončení a tím uvolnění operační paměti by mělo být přenecháno operačnímu systému.

Kapitola 4

Návrh aplikace

4.1 Architektura

Architektura aplikace bude typu klient–server a bude centralizovaná. Veškerý řídící i datový provoz bude přes jeden centrální server. Architektura bude podobná komunikační síti ICQ, nikoliv distribuované síti XMPP[1]. Pro centrální server jsem se rozhodl z důvodu jednoduššího adresování zasílaných zpráv, správy uživatelů a práce s certifikáty.

4.2 Zajištění důvěrnosti zpráv

Pro zabezpečení komunikace jsem se rozhodl využít služeb protokolu SSL spolu s klientskými certifikáty. K realizaci využiji balík `javax.net.ssl`, který je dostupný jak pro Java SE, tak pro Javu pro platformu OS Android.

Certifikát serveru bude podepsán sám sebou (self-signed) a pro jeho vygenerování bude využit nástroj `keytool` z Java JDK[2]. Stejným nástrojem bude vygenerován také klientský certifikát, který bude podepsán certifikátem serveru. Serverový certifikát bude uložen do repozitáře „Java KeyStore“[3], do kterého bude umístěn také veřejný certifikát klienta.

Do repozitáře, který bude součástí klientské aplikace bude umístěn veřejný certifikát serveru a klientský certifikát. Operační systém Android však bohužel nepodporuje repozitář typu JKS a proto je nutné ho převést na typ BKS[4], k čemuž slouží např. nástroj Portecle[5]. Při tvorbě SSL komunikace jsem postupoval dle návodu Dr. Herong Yanga uveřejněného v [26].

4.3 Komunikační protokol

Při návrhu komunikačního protokolu jsem se inspiroval protokolem HTTP[6]. Výsledný protokol je taktéž textový, každá zpráva je uvozena řádkem, který udává typ zprávy a tím pádem i obsah následujících políček.

Na každou zaslanou zprávu budou existovat minimálně dvě odpovědi:

[1] Extensible Messaging and Presence Protocol, dříve známé pod označením Jabber, URL <http://www.xmpp.org/>

[2] URL <http://docs.oracle.com/javase/1.3/docs/tooldocs/win32/keytool.html>

[3] URL <http://docs.oracle.com/javase/6/docs/api/java/security/KeyStore.html>

[4] BouncyCastle KeyStore

[5] URL <http://portecle.sourceforge.net/>

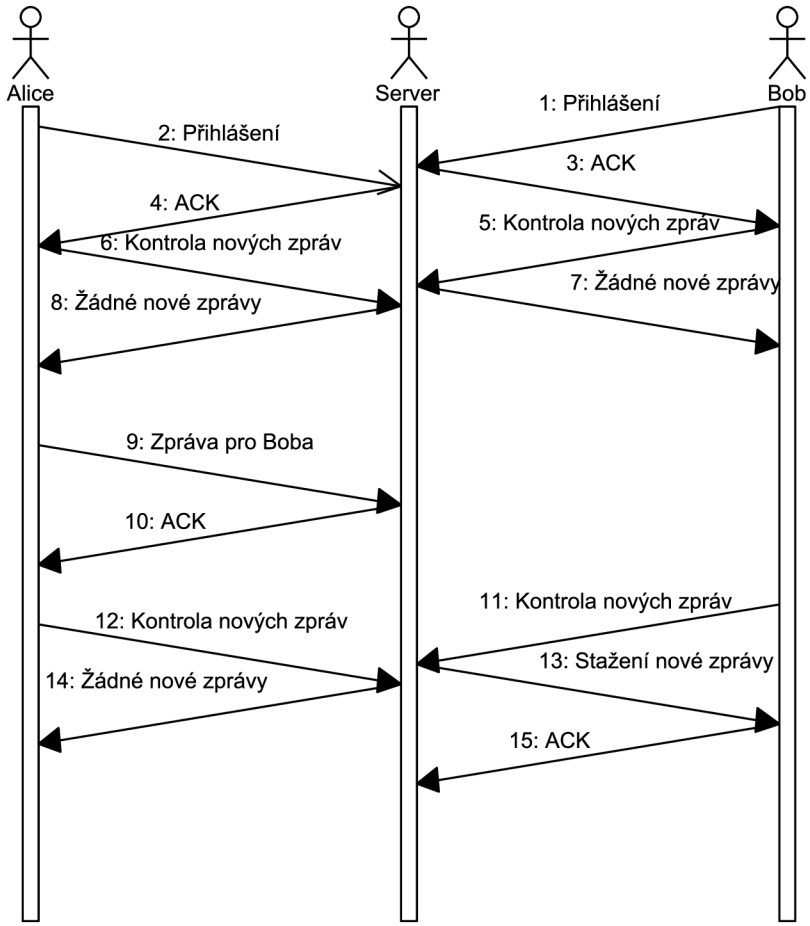
[6] HyperText Transfer Protocol, internetový protokol, URL <http://tools.ietf.org/html/rfc2616>

ACK Odpověď potvrzující přijetí zprávy, resp. správnost údajů, které zpráva obsahuje – např. při přihlašování.

ERR Tato odpověď značí, že při zpracování příchozí zprávy došlo k chybě. Součásti této zprávy vždy bude řádek s chybou, kterou je možno zobrazit uživateli aplikace.

Pro zprávu, která se dotazuje na nové zprávy bude existovat ještě třetí odpověď značící, že na serveru nejsou žádné nové zprávy.

Pomocí sekvenčního diagramu jsem znázornil, jakým způsobem bude probíhat přihlášení uživatele a výměna zprávy mezi dvěma uživateli. Diagram je na obrázku 4.1.



Obrázek 4.1: Sekvenční diagram znázorňující komunikaci dvou osob – Alice a Boba

4.3.1 Odeslání zprávy

Uživatel s ID[7] `id1` zasílá uživateli s ID `id2` zprávu s textem „Ahoj“. Zpráva bude odeslána 24. června 2013 ve 14:14:30 a bude smazána po 30 sekundách od jejího zobrazení. Hodnota `Delete-When-Read` značící, zda má být zpráva smazána ihned po přečtení může nabývat hodnot 0/1 a její použití je výlučné s hodnotou `Delete-After`.

[7] Jednoznačný identifikátor uživatele v serverové databázi

```
MSG
From: id1
To: id2
Delete-When-Read: 0
Delete-After: 30
Sent-Time: 2013-06-24 14:14:30
Text: Ahoj
```

4.3.2 Získání GPS pozice

Uživatel zasílá zprávu s žádostí o poslední pozici uživatele s ID `id`.

```
GET POSITION
User-ID: id
```

Server odpovídá zasláním pozice a času, kdy byla pozice získána.

```
GET POSITION ACK
49.123, 17.456
2013-02-01 16:03:48
```

4.3.3 Registrace nového uživatele „Fiťák“

```
REGISTRATION
Login: Fiťák
Pass-hash: 6E017B5464F820A6C1BB5E9F6D711A667A80D8EA
E-mail: xjerab14@stud.fit.vutbr.cz
```

Odpovědí serveru je – v případě správných hodnot – následující zpráva obsahující ID nového uživatele.

```
REGISTRATION ACK
User-ID: id
```

4.4 Diagram případů užití

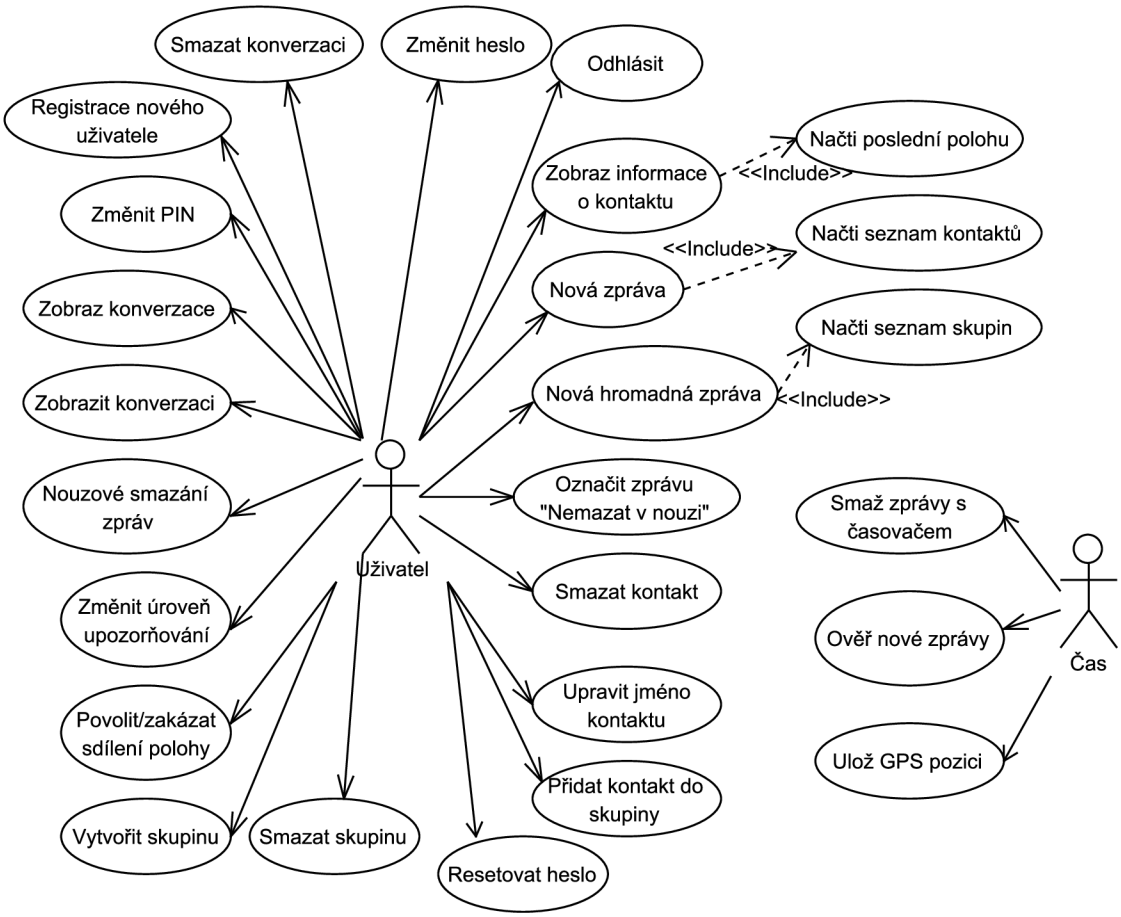
Na základě sepsaných požadavků na klientskou aplikaci v předchozí kapitole jsem vytvořil diagram případů užití, který postihuje funkcionalitu klientské mobilní aplikace. Diagram je vyobrazen na obrázku 4.2. Jsou zde dva účastníci, uživatel a čas.

4.5 Entity-relationship diagramy

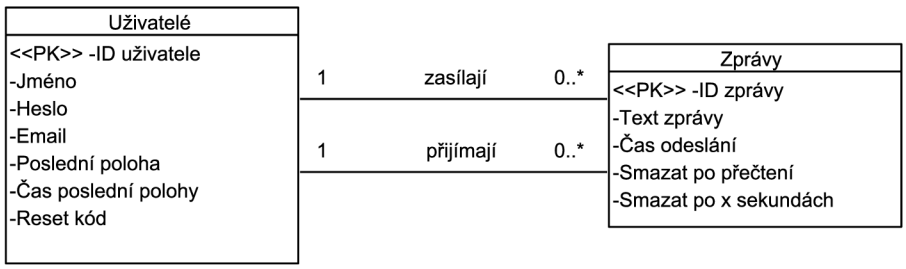
V předchozí části zmíněný diagram případů užití sloužil jako podklad pro tvorbu Entity-relationship diagramů. Ty se využívají pro konceptuální zobrazení dat. Na serveru bude uložen seznam všech uživatelů a nevyzvednuté zprávy.

V klientské aplikaci budou uloženy skupiny, kontakty a zprávy. S ohledem na to, že bude umožněno, aby se do aplikace přihlašovali dva a více uživatelé, musí být v databázi rozlišeno, kterému uživateli co patří. Toho je docíleno atributem „vlastník“, který obsahuje ID uživatele.

Diagramy jsou vyobrazeny na obrázku 4.3 a 4.4.



Obrázek 4.2: Diagram případů užití



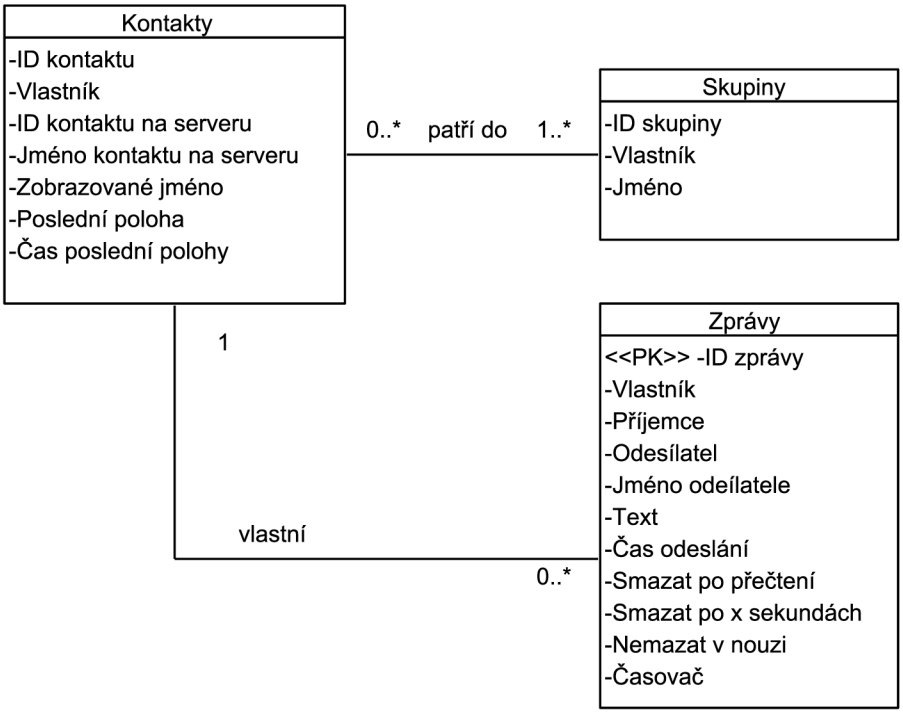
Obrázek 4.3: ER diagram serverové části aplikace

4.6 Ovládání aplikace

4.6.1 Spuštění aplikace

Při prvním spuštění aplikace po instalaci bude uživatel vyzván ke zvolení PIN kódu. Ten bude uložen v hashované podobě do zařízení a bude vyžadován při každém dalším spuštění. Bez zvolení nového PIN kódu nebude možné aplikaci dále spustit.

Při dalších spuštěních bude vyžadován PIN kód, jak je ukázáno na obrázku B.3 a poté bude uživatel přesměrován rovnou do hlavního menu, pokud zůstal přihlášen, tj. neopustil



Obrázek 4.4: ER diagram klientské části aplikace

aplikaci přes tlačítko „Odhlásit“.

4.6.2 Hlavní obrazovka

Hlavní obrazovka aplikace bude obsahovat rozcestník, tedy ikony směřující do adresáře, zpráv, nastavení a na odhlášení z aplikace. Odkazy budou využívat pouze piktogramů bez textového popisu, jak je ukázáno na obrázku 4.5.

4.6.3 Adresář

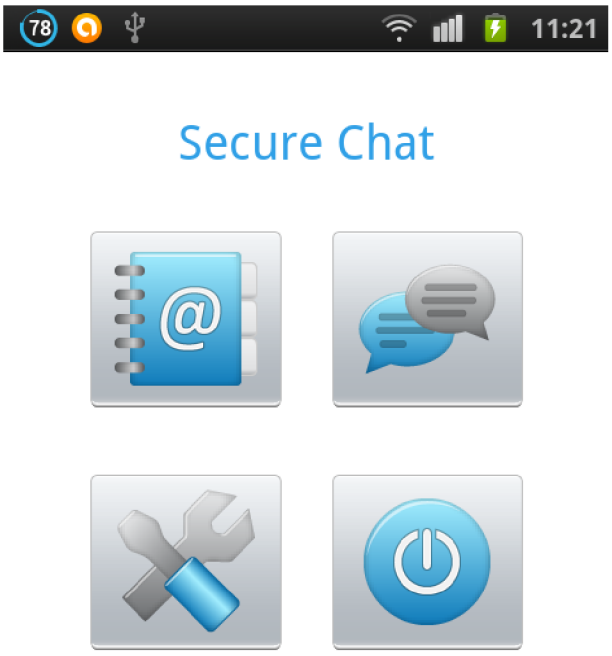
Po otevření adresáře bude k dispozici seznam skupin seřazených dle abecedy vzestupně. Přes menu bude možnost vytvořit novou skupinu.

Po otevření skupiny bude v horní části obrazovky název otevřené skupiny následovaný abecedně seřazeným seznamem kontaktů náležících do této skupiny, jak je ukázáno na obrázku B.6. Menu bude obsahovat celkem tři položky: přidání nového kontaktu, smazání skupiny a zaslání skupinové zprávy. Po dlouhém stisku nad kontaktem bude k dispozici kontextové menu s možností jeho smazání ze skupiny.

Kontakty se budou přidávat zadáním jejich přihlašovacího jména, následně proběhne vyhledání na serveru, pokud takový účet existuje, bude serverem zasláno jeho ID a tento bude přidán do databáze kontaktů a zařazen do skupiny. V případě, že kontakt již figuruje v jiné skupině, bude pouze přidán do pomocné databázové tabulky.

Při mazání kontaktů i skupin bude uživatel dotázán dialogem, zda chce skutečně odstranit požadované položky. Při mazání kontaktu dojde ke smazání přidružených zpráv, a to pouze v případě, že daný kontakt se již nevyskytuje v žádné jiné skupině.

Po kliknutí na kontakt se zobrazí obrazovka s jeho informacemi. Budou zde tři tlačítka,



Obrázek 4.5: Hlavní menu aplikace

první pro změnu zobrazovaného jméno, druhé pro zaslání zprávy a třetí pro smazání kontaktu. Největší část obrazovky bude zabírat mapa se zobrazenou poslední známou pozicí. Při zobrazení detailu kontaktu dojde – v případě připojení k Internetu – vždy k pokusu o získání aktuální polohy ze serveru. Pokud nebude k dipozici žádná informace o poloze kontaktu, bude mapa vycentrována na střed České republiky a uživatel bude na tuto skutečnost upozorněn.

4.6.4 Konverzace

Uspořádání konverzací by mělo být totožné s uspořádáním SMS zpráv na telefonech s OS Android. Konverzace budou seřazeny sestupně dle času poslední zprávy v konverzaci a jednotlivé zprávy v konverzaci poté vzestupně dle času, jak je vidět na obrázcích B.4 a B.1

Nad seznamem konverzací budou dvě tlačítka, jedno pro napsání nové individuální zprávy, druhé pro skupinovou zprávu.

Odchozí zprávy by měly být barevně a zarovnáním odlišeny od příchozích, opět identickým způsobem jako SMS zprávy, tedy odchozí zpráva bude zarovnaná vpravo, příchozí vlevo.

U zprávy bude napsán čas odeslání a v případě, že se bude jednat o zprávu, která má být smazána po přečtení či po uplynutí nastaveného času, bude u ní zobrazena ikonka stopek.

Pod zprávami budou k dispozici dvě tlačítka, jedno pro vytvoření odpovědi – nové zprávy, druhé pro smazání celé konverzace. Umístění ve spodní části displeje volím kvůli snadné dostupnosti tohoto tlačítka palcem pravé ruky při držení přístroje v pravé ruce.

Aplikace bude pochopitelně umožňovat i zaslání zprávy osobě, která nemá odesílatele mezi svými kontakty. Pro tyto případy bude nutné doplnit přijímané zprávy o přihlašovací

jméno odesílatele. Při zobrazení takové konverzace nebude k dispozici tlačítko pro odpověď. Nejprve bude nutné uživatele zařadit mezi své kontakty.

Při dlouhém stisku nad zprávou se zobrazí kontextové menu s možností smazání jednotlivé zprávy či nastavení, resp. odebrání příznaku „Nemazat v nouzi“.

Tvorba a odesílání zprávy

Jak již bylo zmíněno v požadavcích, veškeré možnosti nastavení týkající se zprávy budou na jedné obrazovce. Výběr příjemců bude omezen na seznam kontaktů, resp. skupin z adresáře. Vybrat bude možné pouze jeden kontakt, resp. skupinu.

Bude existovat možnost nastavit čas odeslání zprávy, počet sekund od přečtení, po kterých bude zpráva smazána a zda má být smazána okamžitě po přečtení.

V případě, že uživatel nebude připojen k Internetu, bude odesílaná zpráva uložena do lokální databáze a aplikace se ji bude pokoušet odeslat při kontrole nových zpráv.

4.6.5 Ukončení aplikace

Aplikace bude umožňovat dva způsoby ukončení. První bude pomocí hardwarového tlačítka „zpět“ na telefonu a znamená pouze odebrání aplikace z popředí. Uživatel bude stále přihlášen, upozorňován na nové zprávy a kdykoliv bude moci aplikaci opětovně otevřít a po zadání správného PIN kódu bude v hlavním menu.

Druhý způsob ukončení bude prostřednictvím tlačítka „Odhlásit“ z hlavního menu aplikaci a bude znamenat ukončení veškeré činnosti aplikace v telefonu a pochopitelně také odhlášení uživatele z ní. Při následném spuštění bude kromě PIN kódu vyžadováno zadání uživatelského jména a hesla.

4.7 Serverová část

Serverová část aplikace bude běžet v nepřetržitém provozu na předem dané IP adrese a portu a bude konkurentní [8]. Server bude pouze pasivně čekat na spojení.

4.7.1 Práce se zprávami

Po odeslání zprávy ze zařízení je tato přijata na serveru a uložena do databáze, kde čeká do té doby, než je vyzvednuta jejím příjemcem. Okamžitě po doručení zprávy jejímu příjemci je zpráva vymazána z databáze.

Hromadná zpráva bude zasílána z klienta jako jeden celek s uvedením všech příjemců a až na serveru dojde k jejímu rozdělení na jednotlivé zprávy a takto budou uloženy do databáze a připraveny pro příjemce k vyzvednutí.

4.7.2 Práce s uživateli

Databáze serveru bude obsahovat tabulku se seznamem uživatelů, která bude obsahovat především ID uživatele, jeho přihlašovací jméno a SHA-2 otisk hesla. Oproti těmto údajům bude prováděna autentizace uživatele při přihlašování.

[8] Konkurentní server dokáže obsluhovat více požadavků naráz, typicky vytvořením nového procesu/vlákna při novém příchozím požadavku.

Dále bude v databázi uložena poslední pozice a čas jejího získání. V případě zapomenutí hesla bude využita e-mailová adresa uživatele, na kterou bude zaslán ověřovací kód pro vygenerování nového heslo.

Na serveru nebudou ukládány uživatelovy kontakty ani skupiny, toto bude ponecháno v režii klientské aplikace.

Kapitola 5

Implementace

5.1 Použité technologie

Má volba operačního systému Android pro tvorbu této aplikace byla dána především tím, že sám vlastním zařízení s tímto operačním systémem. Dalším důvodem bylo velké rozšíření této platformy mezi koncovými uživateli. Podle analytické firmy comScore, Inc.[1] dosahovaly přístroje s OS Android ve Spojených státech amerických v měsíci březnu roku 2013 zastoupení 51,7 % mezi chytrými telefony. Na druhém místě se nacházejí zařízení firmy Apple, Inc. s podílem 38,9 % [22].

Volbou mobilního operačního systému je dán také programovací jazyk Java, byť existuje i možnost vyvíjet v jazyce C/C++. Java je však oficiálně podporována firmou Google [23].

U serverové části aplikace jsem měl volnější ruku jak ve volbě operačního systému, tak i programovacích jazyků. Rozhodl jsem se využít taktéž programovacího jazyk Java, konkrétně edici Java SE[2] a to především z toho důvodu, že využívá stejné knihovny jako Java použitá v operačním systému Android. Dalším důvodem byla přenositelnost výsledné aplikace, budoucnost serverové části aplikace je tedy otevřená více řešením.

Databáze serverové části nebude nikterak rozsáhlá a já se rozhodl využít služeb databázového systému MySQL, hlavním důvodem pro toto rozhodnutí byla má dobrá znalost tohoto systému z jiných projektů.

5.1.1 Java

Programovací jazyk Java je objektově orientovaný, navržený Jamesem Goslingem, vyvinutý firmou Sun Microsystems a představený v roce 1995. Od roku 2007 je Java vyvíjena jako otevřený projekt pod licencí GNU GPLv2 [24]. Hlavní předností, kterou se Java vyznačuje, je její přenositelnost. Výsledné aplikace byly dříve kompilovány do mezikódu, který byl poté interpretován libovolným virtuálním strojem Javy (JVM – Java Virtual Machine). Později se začala provádět před spuštěním aplikace kompilace do strojového kódu (JIT - Just In Time), což na jednu stranu urychlilo běh aplikace, ale na druhou stranu výrazně zpomalilo její start. V současné době se využívá služeb tzv. HotSpot kompilace, kdy je mezikód ze začátku interpretován a později na základě statistik z interpretace jsou určité části kódu přeloženy do strojového kódu. Zároveň s tím jsou prováděny další dynamické optimalizace.

Typová kontrola, neboli ověřování datových typů ve zdrojovém kódu, je silná a statická.

[1] Analytics for a Digital World – comScore, Inc., URL <http://www.comscore.com/>

[2] Java Standard Edition, URL <http://www.oracle.com/technetwork/java/javase/overview/index.html>

Silná typová kontrola znamená, že je programátor nucen explicitně uvádět datový typ každé proměnné, se kterou pracuje. Dalším zástupcem takových jazyků je např. jazyk C. Naopak slabě typovaný jazyk je např. PHP[3].

Za statickou typovou kontrolu se označuje kontrola datových typů prováděná při překladu zdrojového kódu. Opět i zde je dalším zástupcem jazyk C. Statická kontrola dokáže zvýšit efektivitu programu, protože nemusí docházet při každém jeho spuštění ke kontrole datových typů. Zástupcem druhé skupiny, dynamicky typovaných jazyků, může být opět jazyk PHP.

Jednou z typických vlastností jazyka Java je jeho práce s pamětí. Programátor se nemusí sám starat o její alokování a především uvolňování, to má na starosti garbage collector, který dokáže automaticky vyhledat již nepoužívané alokované části paměti a ty uvolnit. Díky této vlastnosti jsou programy psané v Javě podstatně méně náchylné na programátorské chyby při práci s pamětí, např. v porovnání s jazykem C. [27]

5.1.2 Databáze MySQL a SQLite

Ve své práci využívám služeb dvou různých relačních systémů řízení báze dat (SŘBD). První z nich, SQLite, je standardní součástí operačního systému Android a tudíž je využit v klientské aplikaci. Oproti tomu, systém MySQL je využit na serverové části aplikace.

Databáze SQLite

SQLite je relativně malý databázový systém, který je napsán v jazyce C Dwaynem Richardem Hippem. Velikost výsledné knihovny je menší než 350 kB [29], v závislosti na cílové platformě a použitých rozšířeních může být menší, i větší.

Hlavní rozdíl oproti databázovým systémům založeným na principu klient–server, které běží jako samostatný proces, je SQLite pouze knihovna, která se připojí k výslednému programu a lze ji okamžitě pomocí jednoduchého rozhraní používat.

I přestože dosahuje SQLite takových malých rozměrů, je v ní implementován téměř celý standard SQL-92. Seznam celkem pěti prvků, které nejsou v SQLite podporovány je uveden v [30]. S ohledem na potřeby pro tento projekt nejsou tato omezení jakkoliv svazujícími. [28]

Výchozí nastavení kódování databáze SQLite je UTF-8, další podporovaná kódování jsou UTF-16BE[4] a UTF-16LE[5]. SQLite nemá takovou pestrou škálu datových typů jako např. MySQL a podporuje pouze následujících pět datových typů:

- **NULL**.
- **INTEGER** Znaménkové číslo uložené na 1 – 8 bytech v závislosti na jeho velikosti.
- **REAL** Číslo s plovoucí řádovou čárkou uložené na 8 bytech.
- **TEXT** Řetězec uložený v jednom ze tří podporovaných kódování, viz výše.
- **BLOB** Binární data.

[31]

[3] PHP: Hypertext Preprocessor, skriptovací jazyk, určený především pro vývoj internetových aplikací.
[4] Big Endian – nejvýznamnější byte je uložen na nejnižší adrese
[5] Little Endian – nejméně významný byte je uložen na nejnižší adrese

Databáze MySQL

Multiplatformní databázový systém MySQL byl vyvinut švédskou firmou MySQL AB, později převzat společností Sun Microsystems, která je dceřinnou společností firmy Oracle Corporation. MySQL je distribuováno pod licencí GNU GPL, resp. komerční licencí.

Komunikace programátora s databází je založena na standardizovaném dotazovacím jazyku SQL[6].

Architektura databáze MySQL je výrazně odlišná od jiných databázových serverů a dá se popsat třemi vrstvami. První z nich obsahuje služby, které obsluhují nástroje komunikace klient–server, které jsou založeny na síti Internet. Druhá vrstva představuje hlavní část databáze, dochází zde k syntaktické analýze a optimalizaci dotazů. Také jsou zde kódy zabudovaných funkcí. Třetí vrstva obsahuje úložiště dat (storage engine), která se starají o ukládání a získávání požadovaných dat. [32]

Ve své práci jsem zvolil pro ukládání dat úložiště `InnoDB`, které – na rozdíl od úložiště `MyISAM` – podporuje cizí klíče.

5.2 Použité knihovny

Ve své práci jsem využil několik externích knihoven, které nejsou standardní součástí používaných technologií.

mysql-connector-java-5.1.24-bin.jar
: API definující rozhraní pro přístup k databázi MySQL v Javě[7].

mail.jar
: API pro tvorbu a zasílání e-mailů v Javě[8].

Java Cryptography Extension Unlimited Strength Jurisdiction Policy Files
: Rozšíření Javy umožňující konverzi JKS do formátu BKS[9].

android-support-v4.jar
: Knihovna obsahující třídy podporující vývoj se zpětnou kompabilitou na OS Android[10].

google-play-services.jar
: Knihovna umožňující využívání služeb Google Map[11].

5.3 Ukládání aplikačních dat

Operační systém Android nabízí vývojářům celkem tři způsoby, jak v aplikaci ukládat data. Každý z nich je určen pro jiný typ dat, ve své aplikaci využívám dva z nich. [33]

[6] Structured Query Language
[7] URL <http://dev.mysql.com/downloads/connector/j/>
[8] URL <https://java.net/projects/javamail/pages/Home>
[9] URL <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>
[10] http://developer.android.com/tools/extras/support-library.html
[11] URL <http://developer.android.com/google/play-services/index.html>

5.3.1 Lokální databáze SQLite

Databáze obsahuje celkem pět tabulek, do kterých jsou ukládány skupiny, kontakty a zprávy. Dvě tabulky jsou pomocné, jedna slouží k zařazování kontaktů do více skupin, druhá pro ukládání zpráv, které není možné momentálně odeslat.

Komunikace s databází se děje prostřednictvím třídy `MyDbHelper`, která rozšiřuje třídu `SQLiteOpenHelper`. V této třídě je definována struktura všech tabulek, aktuální verze databáze, název databáze a operace, které se mají provést při tvorbě a aktualizaci struktury databáze.

Tabulka kontaktů

Struktura tabulky včetně popisu jednotlivých sloupců je uvedena v tabulce 5.1. Při přidání nového kontaktu jsou hodnoty sloupců `contact_login` a `display_name` stejné, sloupce obsahující informace o poloze mají hodnotu `NULL`. Údaj o vlastníkovi kontaktu je nutné evidovat z důvodu možnosti přihlášení více uživatelů v jednom zařízení. Datum a čas získání poslední polohy kontaktu ukládám jako řetězec, protože SQLite nepodporuje datový pro práci s časem a ukládání jako `INTEGER` by vedlo k větší režii při převodu do lidsky čitelné podoby a zpět.

Sloupec	**Typ**	**Popis**
_id	INTEGER	primární klíč tabulky
owner	INTEGER	ID uživatele, který vlastní tento kontakt
contact_id	INTEGER	ID kontaktu v serverové databázi
contact_login	TEXT	přihlašovací jméno kontaktu
display_name	TEXT	zobrazované jméno v aplikaci
last_position	TEXT	poslední pozice kontaktu
time	TEXT	čas poslední pozice

Tabulka 5.1: Struktura tabulky kontaktů

Součástí této tabulky jsou dva unikátní klíče, které zajišťují, že jeden kontakt nebude přidán vícekrát a že nedojde k přejmenování dvou kontaktů na stejné jméno. Pokud k takovému pokusu dojde, bude databází vyvolána výjimka. Tyto klíče jsou ve schématu databáze zapsány pomocí příkazu `UNIQUE(owner, contact_id) ON CONFLICT ABORT`,
resp. `UNIQUE(owner, display_name) ON CONFLICT ABORT`.

Tabulka zpráv

Do tabulky zpráv, která je detailně popsána v tabulce 5.2, jsou ukládány jak příchozí, tak odchozí zprávy. Opět je zde nutné evidovat majitele zprávy. Sloupec `sender_login` je určen pro příchozí zprávy, které mohou být odeslány uživatelem, který není mezi kontakty příjemce, jak je uvedeno v 4.6.4.

U této tabulky se nabízí použití cizích klíčů u sloupce s ID odesílatele, resp. příjemce. To však není možné z toho důvodu, že vždy právě jeden z těchto sloupců bude obsahovat ID přihlášeného uživatele, který nebude mezi kontakty.

Význam sloupce `timer_set` je podrobně popsán v části 5.4.4.

Sloupec	Typ	Popis
_id	INTEGER	primární klíč tabulky
owner	INTEGER	ID uživatele, který vlastní tuto zprávu
sender	INTEGER	ID odesílatele
sender_login	TEXT	přihlašovací jméno odesílatele
receiver	INTEGER	ID příjemce
text	TEXT	text zprávy
sent_time	TEXT	datum a čas odeslání zprávy
timer_set	INTEGER	je nastaven časovač pro tuto zprávu? (viz 5.4.4)
delete_when_read	INTEGER	smazat po přečtení?
delete_after	INTEGER	smazat po x sekundách
dont_delete	INTEGER	nemazat v nouzi

Tabulka 5.2: Struktura tabulky zpráv

Tabulka zpráv k odeslání

Pokud není uživatel připojen k síti Internet a pokouší se odeslat zprávu, je tato uložena do tabulky `to_send`. Její struktura je vidět v tabulce 5.3. S periodickou kontrolou nových zpráv na serveru dochází také ke kontrole obsahu této tabulky. Pokud obsahuje nenulový počet položek a uživatel je již ve stavu online, jsou zprávy odeslány a vymazány z této tabulky.

Je zde využito příznaku `is_group`, který značí, zda se jedná o skupinovou zprávu či nikoliv. Pokud ano, je ve sloupci `receiver` uloženo pole obsahující ID příjemců. Z toho důvodu je datový typ sloupce `TEXT`, nikoliv `INTEGER`.

Sloupec	Typ	Popis
_id	INTEGER	primární klíč tabulky
is_group	INTEGER	příznak, zda se jedná o skupinovou zprávu
sender	INTEGER	ID odesílatele
receiver	TEXT	ID příjemce/příjemců
text	TEXT	text zprávy
sent_time	TEXT	datum a čas odeslání zprávy
delete_when_read	INTEGER	smazat po přečtení?
delete_after	INTEGER	smazat po x sekundách

Tabulka 5.3: Struktura tabulky zpráv, které čekají na odeslání

Tabulka skupin

Tabulka se skupinami je velmi jednoduchá, opět i zde je nutné uvádět vlastníka skupiny. Aby bylo zabráněno vytvoření dvou stejně pojmenovaných skupin je zde unikátní klíč zapsaný `UNIQUE(owner, title) ON CONFLICT ABORT`. Popis této databázové tabulky je vidět v tabulce 5.4.

Sloupec	Typ	Popis
_id	INTEGER	primární klíč tabulky
owner	INTEGER	ID vlastníka skupiny
title	TEXT	název skupiny

Tabulka 5.4: Struktura tabulky skupin

Tabulka kontaktů a skupin

Pomocná tabulka `contacts_groups` popsána tabulkou 5.5 slouží k realizaci vztahu 1:N mezi kontakty a skupinami. I zde se nabízí použití cizích klíčů, bohužel podpora pro cizí klíče je v SQLite až od verze 3.6.19 [34], která je dodávána oficiálně až s Androidem verze 2.2. Z důvodu zachování možnosti zpětné kompatibility i pro starší zařízení či zařízení, u kterých se výrobce rozhodl zvolit nižší verzi databáze, jsem od použití cizích klíčů upustil a integritu databáze zajišťuji programově.

Přidání jednoho kontaktu vícekrát do jedné skupiny zabraňuje použití unikátního klíče `UNIQUE(contact_id, group_id) ON CONFLICT ABORT`.

Sloupec	Typ	Popis
contact_id	INTEGER	ID kontaktu
group_id	INTEGER	ID skupiny

Tabulka 5.5: Struktura tabulky kontaktů a skupin

5.3.2 Ukládání nastavení aplikace

Pro ukládání nastavení aplikace jsem zvolil možnost ukládání typu klíč–hodnota nabízeného API[12] `SharedPreferences`.

Jsou zde ukládány následující informace:

- ID právě přihlášeného uživatele,
- informace o tom, zda je uživatel přihlášen,
- informace o tom, zda již byl nastaven PIN kód při prvním spuštěním aplikace,
- potvrzovací kód pro resetování hesla uživatele,
- uživatelské nastavení:
  - PIN kód,
  - falešný PIN kód,
  - povolení/zakázání sdílení GPS polohy,
  - povolení/zakázání upozorňování na nové zprávy,
  - povolení/zakázání pouze skrytého upozorňování,

[12] Application Programming Interface

- zvuk nové zprávy,
- povolení/zakázání vibrací,
- povolení/zakázání ukládání příchozích zpráv,
- povolení zakázání ukládání odchozích zpráv.

Pro ukládání výše uvedených hodnot je využíváno modifikátoru `MODE_PRIVATE`, který zaručuje, že se k hodnotám nedostane žádná jiná aplikace, jako je tomu v případě zbylých dvou modifikátorů `MODE_WORLD_READABLE` a `MODE_WORLD_WRITEABLE`.

Při získávání konkrétní hodnoty je nutno specifikovat také výchozí hodnotu v druhém parametru metody, která se použije v případě, že pod uvedeným klíčem nebude žádná hodnota existovat. Tohoto se využívá kupříkladu pro detekci prvního spuštění aplikace, jak je uvedeno níže. V tomto případě dojde k zobrazení výzvy nastavení PIN kódu. Po jeho nastavení je hodnota klíče `first_run` nastavena na `true`.

```
getSharedPreferences(SYSTEM_PREF, MODE_PRIVATE).getBoolean("first_run", false);
```

5.3.3 Ukládání souborů

OS Android nabízí také možnost ukládat libovolná data, této možnosti není v aplikaci využito.

5.4 Bezpečnostní prvky aplikace

5.4.1 Hashovací funkce a kryptografická sůl

Bylo by velmi nezodpovědné ukládat hesla uživatelů v otevřené – nezašifrované – podobě, k těmto účelům slouží jednosměrné funkce. Jednosměrná funkce je taková, pro kterou je jednoduché spočítat $y = f(x)$ za zadaného x, ale spočítat z výsledného y zpět hodnotu x je velmi časově náročné. Jednoduchým příkladem takové funkce je součin dvou čísel.

V informačních systémech se běžně používají kryptografické hashovací algoritmy, mezi nejznámější patří MD5[13] a SHA[14]. Algoritmus MD5 je již nedoporučovaný. Nejprve byl prolomen pouze teoreticky, posléze také prakticky a v roce 2005 publikoval Vlastimil Klíma příspěvek, ve kterém popisuje metodu nalézání kolizí na tehdy standardním notebooku v řádu hodin [35]. Nástupcem algoritmu MD5 je rodina hashovacích funkcí SHA, které navrhuje Národní bezpečnostní agentura ve Spojených státech amerických. Tento algoritmus se vyskytuje ve verzích SHA-0, SHA-1, SHA-2 a nově SHA-3. U verze SHA-1 byly nalezeny v roce 2005 nedostatky a není oficiálně schválená pro používání po roce 2010 [36]. Verze SHA-2 je prozatím považována za bezpečnou a z tohoto důvodu jsem se ji rozhodl použít ve své práci. [37]

Abych eliminoval riziko úspěšného útoku hrubou silou nebo za pomoci duhových tabulek na uložené otisky hesel, rozhodl jsem se k heslu přidávat kryptografickou sůl. Ta je tvořena dvěma řetězci, které obsahují pseudonáhodně vygenerované alfanumerické a speciální znaky o celkové délce 26 znaků. Při nastavené minimální délce hesla na šest znaků je tedy výsledně hashováno celkem 32 znaků.

[13] Message-Digest algorithm
[14] Secure Hash Algorithm

5.4.2 Skrytí aplikace v zařízení

Každá aplikace pro OS Android musí obsahovat soubor `AndroidManifest.xml`. V tomto souboru jsou obsaženy informace o aplikaci, jsou zde definována oprávnění a také vstupní bod aplikace. Pokud však vstupní bod aplikace chybí, není tato zobrazeno v seznamu aplikací a také ji nelze běžným způsobem spustit.

V souboru `AndroidManifest.xml` lze také zaregistrovat přijímač všesměrového vysílání (broadcast receiver), který bude reagovat na odchozí hovory:

```
<receiver android:name="cz.jerabekjakub.securechat.OutgoingCallReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.NEW_OUTGOING_CALL" />
    </intent-filter>
</receiver>
```

Na prvním řádku je uvedena třída, která bude bude obsluhovat přijaté vysílání a na třetím řádku je uveden typ vysílání, na který bude tento přijímač reagovat – v tomto případě na veškeré odchozí hovory ze zařízení.

Vlastní třída `OutgoingCallReceiver` musí rozšiřovat třídu `BroadcastReceiver` a povinně implementovat metodu `onReceive()`. Na následujícím kódu je vidět, jak dochází ke spuštění aplikace poté, co je získáno volané číslo a to porovnáno s číslem „7890“. Důležitý je také poslední řádek, který ukončuje odchozí hovor a zabraňuje tak pokračování hovoru po ukončení aplikace.

Pro zachycení odchozích hovorů je nutné vlastnit oprávnění `PROCESS_OUTGOING_CALLS`.

```
String phoneNumber = intent.getExtras().getString(Intent.EXTRA_PHONE_NUMBER);

if (phoneNumber.equals("7890")) {
    Intent intent1 = new Intent(context, PINActivity.class);
    intent1.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(intent1);
    setResultData(null);
}
```

5.4.3 Ochrana proti SQL injection

SQL injection je útok na databázovou vrstvu, při kterém útočník modifikuje vkládaná data takovým způsobem, aby došlo k narušení vykonávaného příkazu. Typickým příkladem je vložení uvozovky do řetězce což způsobí při jeho vložení do databázového dotazu jeho předčasné ukončení. Následovat může útočníkům dotaz, který například způsobí vyprázdnění celé databáze.

V klientské aplikaci jsou hodnoty do databáze vkládány pomocí speciální metody `insertOrThrow`, která je součásti Android API a automaticky vkládané hodnoty ošetřuje. Na následujícím zdrojovém kódu, který přidává novou skupinu, je ukázáno její použití včetně zachycení případných výjimek. Pro aktualizaci dat v databázi je využívána metoda `update`, která pracuje na stejném principu

```
ContentValues values = new ContentValues();
values.put("owner", get_user_id());
values.put("title", strTitle);
```

```
try {
    database.insertOrThrow(MyDbHelper.TABLE_GROUPS, null, values);
} catch (SQLiteConstraintException ex) {
    Toast.makeText(getApplicationContext(), R.string.group_title_exists,
          Toast.LENGTH_LONG).show();
} catch (SQLException ex) {
    Toast.makeText(getApplicationContext(), R.string.general_db_error,
          Toast.LENGTH_LONG).show();
}
```

U serverové části aplikace je také nutné ošetřovat vkládaná data a pro toto jsou opět využívány možnosti jazyka Java. Na níže uvedené ukázce je vkládán nový uživatel do databáze, je zde využito zástupného symbolu `?` při vytváření parametrizovaného SQL dotazu. Hodnoty jsou následně vloženy za použití metody odpovídající datovému typu vkládaných dat.

```
stmt = conn.prepareStatement("INSERT INTO users (name, password, mail)" +
            "values (?, ?, ?)", Statement.RETURN_GENERATED_KEYS);
stmt.setString(1, login);
stmt.setString(2, hash);
stmt.setString(3, mail);
stmt.executeUpdate();
```

5.4.4 Mazání zpráv po uplynutí nastaveného času

Při tvorbě nové zprávy je možné nastavit počet sekund, za které bude zpráva vymazána z příjemcova zařízení od jejího zobrazení. Tvorba nové zprávy je vidět na obrázku 5.1.

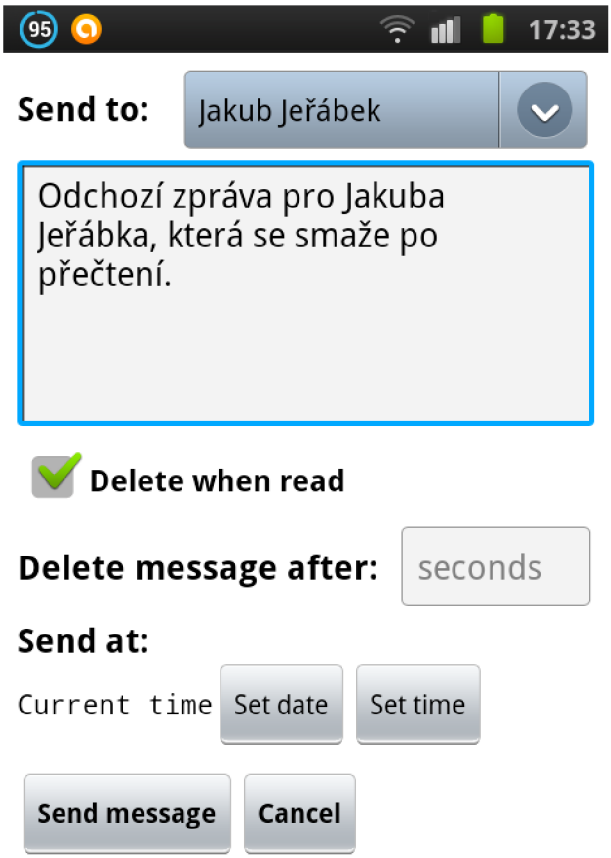
Programově je nastavení mazání zprávy řešeno ve třídě `MyConversationAdapter`, která má za úkol transformovat data vrácená SQL dotazem do grafického prvku `ListView` a kde dochází k nastavení časovače, který slouží k zaslání předem nastaveného všesměrového vysílání za určitý čas. Časovač je realizován využitím třídy `AlarmManager` a je nutné ho nastavovat pouze jednou, čehož je docíleno pomocí příznaku `timer_set` v tabulce zpráv (viz tabulka 5.2). Při každém zobrazení konverzace je prováděna kontrola u každé zprávy, zda se má tato smazat po určitém čase, zda ještě nebyl časovač nastaven a zda se jedná o příchozí zprávu. Pokud zpráva vyhovuje těmto třem podmínkám, je nastaven časovač a změněn příznak v tabulce zpráv.

Časovač po uplynutí nastaveného času zašle zprávu všesměrového vysílání, pro jejíž příjem je zaregistrován příjemce `DeleteAfterBroadcastReceiver`. Ten ze zaslané zprávy získá ID zprávy a zprávu smaže z databáze. Mazání se odehrává i tehdy, je-li aplikace ukončena.

5.4.5 Mazání zpráv po přečtení

Mazání zpráv ihned po přečtení je o poznání jednodušší, dochází k němu totiž ihned po načtení a zobrazení zpráv. Mazání se děje prostřednictvím SQL dotazu, který smaže všechny zprávy, které mají nastaven příznak, patří aktuálně přihlášenému uživateli a odesílatel je uživatel z právě zobrazené konverzace. Oproti předchozímu způsobu mazání se toto děje hromadně, nikoliv po jedné.

Oba zmíněné způsoby mazání zpráv na přání odesílatele jsou plně autonomní a příjemce nemá žádnou možnost je zvrátit.



Obrázek 5.1: Tvorba nové zprávy

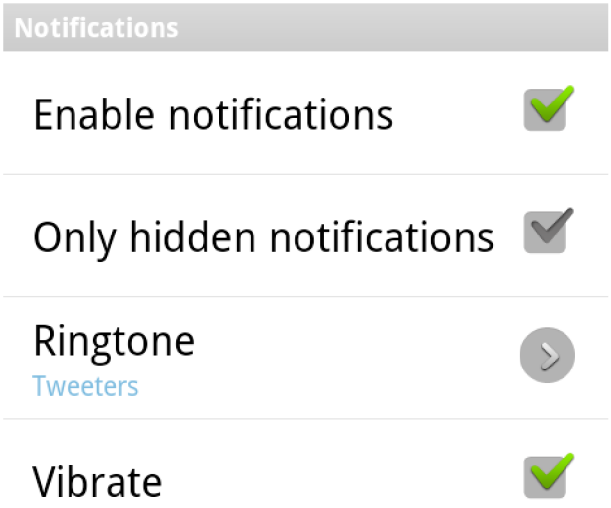
5.5 Upozorňování na nové zprávy

Jak bylo zmíněno v sekci 3.3, někteří uživatelé aplikace by ocenili několik úrovní upozorňování na příchozí zprávy. Jak je ukázáno na obrázku 5.2, aplikace dává k dispozici až pět úrovní upozorňování.

Nejvíce diskrétním je režim s vypnutými notifikacemi. V takovém případě se uživatel o nové zprávě nedozví a je nucen sám kontrolovat konverzace, zda se tam nenachází nová zpráva. Nezatržení políčka „Enable notifications“ způsobí, že se následující tři políčka se stanou neaktivní.

Druhým nejvíce diskrétním režimem jsou skryté notifikace. Ty jsou realizovány pomocí ikony baterie ve stavové liště (obrázek 5.3) a jsou doplněny informací o stavu nabití baterie v procentech (obrázek 5.4). Po kliknutí na upozornění se neotevře aplikace, ale pouze zmizí upozornění. Pro získání informace o baterii je nutné zaregistrovat přijímač všesměrového vysílání pro zprávy typu `ACTION_BATTERY_CHANGED`.

Poslední způsob upozorňování je klasický, tedy ikona aplikace ve stavové liště doprovázená zvoleným zvukem a vibracemi. Po kliknutí na upozornění dojde k otevření aplikace na obrazovce, která vyžaduje zadání PIN kódu. Vibrace lze vypnout, jak je zřejmé z obrázku 5.2 a ve výběru vyzvánění lze zvolit „Ticho“. Upozornění je doplněno o informaci o počtu nových zpráv, jak je vidět na obrázku 5.5. Kombinací zmíněných možností lze tedy dosáhnout dalších tří úrovní upozorňování na příchozí zprávy.



Obrázek 5.2: Možnosti nastavení upozorňování na nové zprávy

Obrázek 5.3: Ikona baterie v levé části upozorňující na novou zprávu

5.6 Vyžadovaná oprávnění aplikace

Klientská aplikace vyžaduje při instalaci schválení následující oprávnění:

Vaše poloha
: Určení přibližné polohy pomocí sítě nebo její upřesnění za pomoci GPS modulu zařízení. Toto oprávnění je vyžadováno, aby bylo možné získat polohu uživatele.

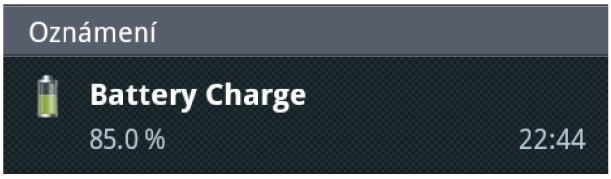
Síťová komunikace
: Plný přístup k internetu a zobrazení stavu sítě. Pro fungování aplikace je nutný přístup k síťovým paketům, také je programově zjišťován stav připojení k síti.

Úložiště
: Přístup k externímu úložišti není nezbytně nutný, slouží pouze pro zvýšení komfortu uživatele. Toto oprávnění je využito při ukládání mapových dlaždic do mezipaměti zařízení a při opakovaném přístupu na detaily kontaktu tak nedochází k jejich stahování.

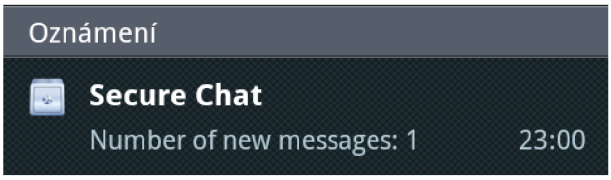
Telefonní hovory
: Díky tomuto oprávnění je možné zachycení odchozích hovorů a při vytočení čísla `7890` spuštění aplikace.

Systémové nástroje
: Zabránění přechodu telefonu do režimu spánku díky čemuž neustane kontrola nových zpráv na serveru.

Řízení hardwaru
: Ovládání vibrací, umožňuje spustit vibrace při příchodu nové zprávy.



Obrázek 5.4: Doplňující informace o stavu nabití baterie



Obrázek 5.5: Standardní upozornění na příchozí zprávu – ikona, název aplikace a počet nových zpráv

5.7 Získávání a zobrazování GPS pozice

Pokud uživatel povolí sdílení své GPS polohy, je tato zjišťována ze dvou zdrojů. První z nich je GPS poskytovatel, který polohu určuje pomocí družic [38]. Získání polohy může chvíli trvat a je nutná přímá viditelnost na satelity. Druhým zdrojem polohy jsou stanice BTS[15] mobilní sítě a přístupové body WiFi sítě [38].

Programově je vše řešeno pomocí třídy `LocationManager` z balíku `android.location`, který je součástí Android API. Nejprve dojde k získání poslední známé polohy od každého zdroje, následně dojde k porovnání časů těchto dvou poloh a novější poloha je uložena. V případě změny polohy dojde k zavolání metody `onLocationChanged`, která zajistí aktualizaci i na serveru, pokud je uživatel online. Získání poslední známé polohy z GPS modulu je ukázáno na následujícím kódu. [39]

```
Location locationGPS = locationManager
                       .getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

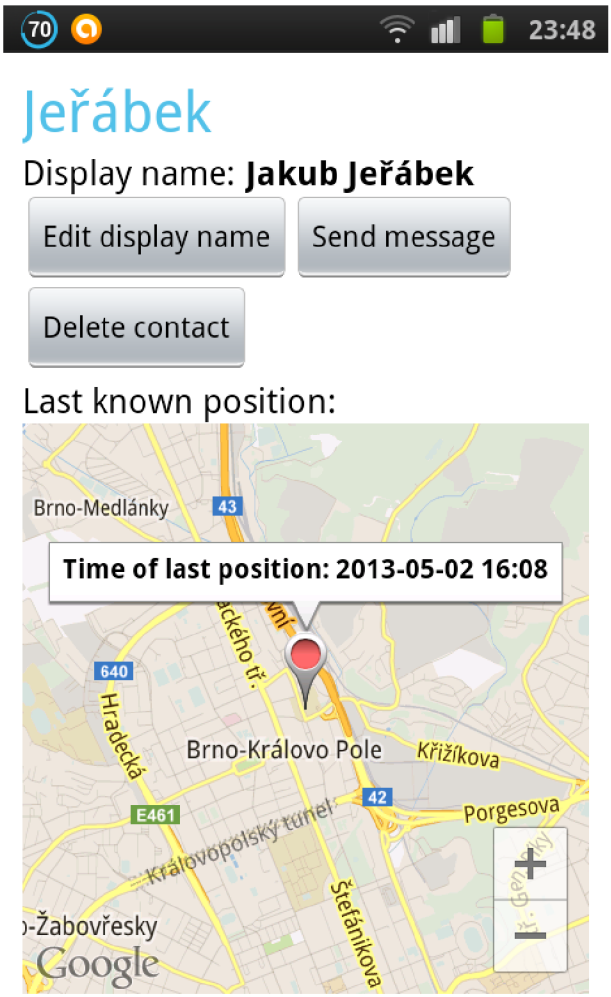
Při zobrazování detailu kontaktu jsou nejdříve získány informace z lokální databáze, včetně poslední známé polohy. Dále je ověřen stav připojení k síti a pokud je uživatel online, je na server odeslána žádost o pozici kontaktu. V případě pozitivní odpovědi je získaná poloha spolu s jejím časem uložena do lokální databáze. Pokud nelze získat polohu ani z lokální, ani ze serverové databáze, je zobrazená mapa oddálena a uživatel o stavu informován. V případě, že poslední známá poloha je z lokální databáze, je na toto uživatel taktéž upozorněn.

Zobrazení značky na mapě včetně časového údaje zajišťuje následující kód. Ukázka je vidět na obrázku 5.6.

```
GoogleMap map = ((SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map)).getMap();

Marker m = map.addMarker(new MarkerOptions().position(position)
              .title(getString(R.string.position) + time));
```

[15] Base Transceiver Station



Obrázek 5.6: Detail uloženého kontaktu včetně zobrazení poslední známé polohy

```
m.showInfoWindow();
map.moveCamera(CameraUpdateFactory.newLatLngZoom(position, POSITION_ZOOM));
```

Aby bylo možné zobrazit mapy od společnosti Google, je nutné nainstalovat do správce Android SDK službu Google Play a tu následně importovat do projektu ve vývojovém prostředí. Dále je nutné si nechat vygenerovat SHA-1 otisk přístupového klíče ke Google API a zaregistrovat se na webu Google APIs Console[16], kde je nutné zapnout službu „Google Maps Android API v2“. Také je zapotřebí zaregistrovat zde svou aplikaci a vygenerovat pro ni přístupový klíč (API Key), který se následně vloží do souboru `AndroidManifest.xml`. [40]

Třídy a jejich metody, které jsem zmínil v předchozích odstavcích využité pro práci s mapou pocházejí z balíku `com.google.android.gms.maps`.

Využitím služeb Google Play přijde vývojář o možnost testovat aplikaci na emulátoru. Spuštění aplikace je možné provádět pouze na fyzickém zařízení [41].

[16] URL <https://code.google.com/apis/console/>

5.8 Serverová část

Princip činnosti serverové části je veskrze jednoduchý. Je zde vytvořen SSL soket a pokud inicializace (SSL handshake) proběhne úspěšně je vytvořeno nové vlákno, které obsluhuje příchozí požadavek.

Ve vlákně je přečten první řádek zprávy obsahující typ požadavku a ten je předán příkazu `switch`. Následuje volání příslušné metody, která zpracuje zbývající část zprávy. Každá metoda vyhazuje několik výjimek a kromě výjimek, které jsou součásti Javy to jsou i vlastní výjimky definované ve třídě `CustomExceptions`. Pro příklad tam patří výjimky `LoginNotFound` a `NoUnreadMessages`, které informují o neexistenci zadaného uživatelského jména a o tom, že na serveru nejsou žádné nové zprávy. Pokud je metodou vyhozena výjimka, je uživatelské aplikace odeslána chybová zpráva s příslušným popisem chyby v lidsky čitelné podobě, aby bylo možné ji přímo zobrazit uživateli.

Při zasílání nových zpráv klientovi je nutné do zprávy také vložit uživatelské jméno odesílatele. To se získává pomocí SQL konstrukce `INNER JOIN` mezi tabulkou zpráv a uživatelů, jak je vidět na níže. V ukázaném dotazu je také vidět, že zde probíhá výběr zpráv, jejichž čas odeslání je aktuální či již uplynul. Tím se zamezuje tomu, aby nebyly odeslány zprávy dříve, než jejich odesílatel nastavil.

```
SELECT m.message_id, m.sender, m.receiver, m.text, m.delete_when_read,
       DATE_FORMAT(m.sent_time, '%Y-%m-%d %H:%i:%s') AS sent_time,
       m.delete_after, u.name
FROM messages m
INNER JOIN users u ON m.sender = u.users_id
WHERE receiver=? AND sent_time<=NOW()
ORDER BY message_id ASC
LIMIT 1
```

5.8.1 Databáze

Serverová část obsahuje pouze dvě tabulky, a to tabulku s uživateli (tabulka 5.6) a zprávami (tabulka 5.7). Tabulka se zprávami je velmi podobná té v klientské části, oproti SQLite jsou zde však použity specifičtější datové typy a cizí klíče do tabulky uživatelů pro sloupce `sender` a `receiver`.

Sloupec	**Typ**	**Popis**
message_id	INT(10)	primární klíč tabulky
sender	INT(10)	ID odesílatele
receiver	INT(10)	ID příjemce
text	TEXT	text zprávy
sent_time	DATETIME	datum a čas odeslání zprávy
delete_when_read	TINYINT(1)	smazat po přečtení?
delete_after	SMALLINT(5)	smazat po x sekundách

Tabulka 5.6: Struktura tabulky zpráv serverové části

Tabulka uživatelů neskrývá žádné překvapení a všechny sloupce jsou téměř samovysvětlující. V této tabulce jsou použity dva unikátní klíče, a to na sloupce `name` a `mail`.

Tím je zabráněno v registraci dvou uživatelů se stejným uživatelským heslem a také použití jedné e-mailové adresy pro více registrací. Poslední tři sloupce tabulky mají výchozí hodnotu `NULL`.

Pokud uživatel zapomene heslo, může požádat o jeho resetování. Na uvedenou e-mailovou adresu je poté zaslán vygenerovaný kód.

Primární klíč tabulky uživatelů je v celé aplikaci používán jako jednoznačný identifikátor jednotlivých uživatelů.

Sloupec	**Typ**	**Popis**
user_id	INT(10)	primární klíč tabulky
name	VARCHAR(30)	ID uživatelské jméno
password	VARCHAR(160)	SHA-2 otisk hesla se solí
mail	VARCHAR(160)	e-mailová adresa
last_position	VARCHAR(30)	poslední známá poloha uživatele
last_position_time	DATETIME	čas poslední známé polohy
reset_code	MEDIUMINT(8)	ověřovací kód pro resetování hesla

Tabulka 5.7: Struktura tabulky kontaktů

5.9 Testování aplikace

Testování aplikace probíhalo v celém průběhu implementace. Serverová část byla nejdříve testována na 64-bitovém operačním systému Windows 7 s nainstalovanou Javou verze 7u13 a databází MySQL verze 5.5.24.

S potřebou testovat aplikaci nejen v lokální síti došlo k přestěhování serverové části na zapůjčený server, na kterém běží operační systém Ubuntu 12.04.2 LTS. Verze databáze MySQL je 5.5.29 a verze Javy 7u15.

Při testování na serverové části jsem využíval ladících výpisů na standardní chybový výstup, pomocí kterých jsem sledoval správnost přicházejících dat z klientské aplikace a zda na ně serverová část správně reaguje.

Klientská aplikace byla testována výhradně na fyzických zařízeních vzhledem k použití služby Google Play, která znemožňuje využití emulátoru. První fyzické zařízení, na kterém byla aplikace otestována byl Samsung Galaxy S s verzí OS Android 2.3.3. Toto zařízení disponuje procesorem o frekvenci 1 GHz, pamětí RAM o velikosti 329,MB a displejem o úhlopříčce 4 palce.

Druhým zařízení byl také telefon z dílem Samsungu, a to Samsung Galaxy Y vybavený OS Android verze 2.3.6. Toto zařízení má pouze tří palcový displej, 832 MHz procesor a 290 MB RAM.

Třetím zařízení byl od výrobce Sony, konkrétně Sony Xperia S s OS Android verze 4.0.4. Dvoujádrový procesor běžel na frekvenci 1,5 GHz, paměti RAM bylo k dispozici 1 GB a úhlopříčka displeje byla 4,3 palce.

Na všech zařízeních se aplikace chovala totožně a její běh byl subjektivně velmi svižný, nepozoroval jsem jakýkoliv rozdíl v rychlostech mezi jednotlivými zařízeními.

Aplikace v zařízení po nainstalování a několik denním používání zabírá 2,67 MB. Její paměťovou náročnost nelze s přesností určit, ale nepravidelným sledováním správce úloh jsem vypozoroval, že v paměti RAM aplikace zabírá mezi 5 – 10 MB.

Velké obavy jsem měl z toho, jak moc bude aplikace vytěžovat baterii při pravidelné kontrole nových zpráv. Tyto obavy se však během dvoudenního testu nepotvrdily. Během dvou dnů nebylo zařízení vůbec nabíjeno a bylo s ním zacházeno standardním způsobem (SMS zprávy, telefonní hovor, e-mail, prohlížení Internetu). Dle statistik využití baterie (Nastavení → O telefonu → Využití baterie) se aplikace na vybití baterie podílela pouze ze dvou procent, což byla shodná hodnota s aplikacemi Gmail či Twitter.

Při testování klientské aplikace jsem taktéž využíval ladících výpisů, které lze zobrazit v konzoli LogCat vývojového prostředí Eclipse[17]. Pro ladění jsem také využíval službu DDMS[18], která je součástí vývojářských nástrojů, které jsou součástí Eclipse [42]. Při testování práce s GPS modulem jsem využíval aplikaci CatLog – Logcat Reader![19], která umožňuje zobrazovat systémový log.

[17] Verze Eclipse se zabudovanou podporou pro vývoj pro OS Android, URL <http://developer.android.com/sdk/index.html>

[18] Dalvik Debug Monitor Service

[19] URL <https://play.google.com/store/apps/details?id=com.nolanlawson.logcat>

Kapitola 6

Závěr

6.1 Přístup ke splnění zadání

Prvním úkolem této bakalářské práce bylo provést analýzu řešení aplikací, které jsou nabízeny v jednotlivých obchodech s mobilními aplikacemi a zaměřují se na zabezpečenou textovou komunikaci. Analýzou jsem zjistil, že existuje několik aplikací, jejichž jedinou úlohou je odeslat zašifrovaný text, bohužel však již neřeší přenos hesla a neposkytují ani komplexní správu kontaktů či zpráv. Další skupinou aplikací jsou již zavedené komunikační aplikace, které nabízejí uživateli spoustu možností při správě kontaktů a zpráv. U těchto aplikací však není zajištění důvěrnosti přenášených zpráv na prvním místě a důvěrnost již doručených zpráv není nijak zajišťována.

Na základě analýzy bylo dalším úkolem specifikovat požadavky, které by měla splňovat aplikace, která se bude zaměřovat na důvěrnost zasílaných zpráv, a to nejen při jejich samotném přenosu. Zároveň by měla aplikace svému uživateli poskytovat dostatečný komfort při správě kontaktů a zpráv. Nejvíce jsem se při návrhu inspiroval pluginem Ekboo (viz 2.2.1). Návrh popisuje jak obecné vlastnosti aplikace, tak i její bezpečnostní funkce. Zabývá se např. několika způsoby, jak zabránit neoprávněnému spuštění aplikace, jak zajistit smazání zprávy ze zařízení příjemce aj.

V návrhu aplikace využívám jazyka UML[1], konkrétně sekvenční diagram pro znázornění komunikace dvou osob, Entity-Relationship diagramy pro konceptuální zobrazení ukládábých dat v klientské i serverové části a diagram případů užití pro popis klientské aplikace. V této části práce jsou také ukázky navrženého textového protokolu a detailní popis uživatelského rozhraní aplikace.

V kapitole věnované implementaci nejdříve popisuji jednotlivé technologie včetně důvodů, proč jsem zvolil právě je. Hlavní část textu se však věnuje detailnímu popisu ukládání dat v obou částech řešení, popisu realizace bezpečnostních funkcí aplikace a další vybrané části aplikace. Součástí implementace bylo také testování na několika fyzických zařízeních, které odhalilo některé nedostatky, které jsem však operativně vyřešil.

6.2 Možný další vývoj aplikace

Do budoucna by bylo vhodné uvažovat o rozšíření aplikace o několik dalších funkcí. Nabízí se mít možnost zjistit, zda zpráva byla doručena a zda byla přečtena včetně času jejího

[1] Unified Modeling Language

přečtení. Další funkcí, kterou mají některé jiné aplikace je možnost smazat na dálku zprávu ze zařízení příjemce.

Logickým vyústěním předchozího snažení je také publikace aplikace v obchodě s aplikacemi Google Play[2], kde by byla podrobena podstatně důkladnějšímu otestování na zařízeních rozlišných výrobců, rozlišení a verzí operačního systému. Vzhledem k tomuto faktu by bylo vhodné doplnit aplikaci o službu BugSense[3], která umožňuje vývojáři sledovat jaké chyby se v jeho aplikaci objevily a na kterém zařízení to bylo.

S dalším rozvojem aplikace a zvýšení její atraktivity a konkurenceschopnosti by bylo možné přidat možnost zasílat soubory. Přidání podpory dalších jazyků by jistě mělo stejný efekt a pro tuto možnost je již aplikace z větší části připravena používáním jazykových souborů.

[2]URL <https://play.google.com/store>
[3]URL <http://www.bugsense.com/>

Literatura

[1] PROCHÁZKA, Martin. Obyčejné mobily končí, Češi si kupují už jen ty chytré. In: Novinky.cz [online]. 2013, 25. dubna 2013 [cit. 2013-05-05]. Dostupné z: http://www.novinky.cz/finance/299853-obycejne-mobily-konci-cesi-si-kupuji-uz-jen-ty-chytre.html

[2] As US Smartphone Penetration Grows, So Does Apple's Market Share. In: MarketingCharts [online]. 2013, March 7, 2013 [cit. 2013-05-06]. Dostupné z: http://www.marketingcharts.com/wp/topics/signs-of-whats-to-come/as-us-smartphone-penetration-grows-so-does-apples-market-share-27600/

[3] ČÍŽEK, Jakub. Ukradli jsme účet k Facebooku. S Androidem za pět sekund. In: Živě.cz [online]. 2011 [cit. 2013-05-08]. Dostupné z: http://www.zive.cz/clanky/ukradli-jsme-ucet-k-facebooku-s-androidem-za-pet-sekund/sc-3-a-157411/default.aspx

[4] MEYER, David. Chat apps have overtaken SMS by message volume, but how big a disaster is that for carriers?. In: GigaOM [online]. 2013, Apr. 29, 2013 [cit. 2013-05-07]. Dostupné z: http://gigaom.com/2013/04/29/chat-apps-have-overtaken-sms-by-message-volume/

[5] Hamish Medlin – Android Software [online]. [2012] [cit. 2013-05-07]. Dostupné z: http://android.hamishmedlin.com/

[6] Crypt Haze. Aplikace pro Android ve službě Google Play [online]. 2012, 1. červenec 2012 [cit. 2013-05-07]. Dostupné z: http://play.google.com/store/apps/details?id=net.rehacktive.cryptdroid

[7] Ekboo Crypto Chat [online]. [2012] [cit. 2013-05-07]. Dostupné z: http://www.ekboo.com/index.php

[8] ChatSecure [online]. 2012 [cit. 2013-05-07]. Dostupné z: http://chrisballinger.info/apps/chatsecure/

[9] NSA Suite B Cryptography. National Security Agency/Central Security Service: Defending Our Nation, Securing the Future. [online]. 2009, [Dec 6, 2012] [cit. 2012-12-07]. Dostupné z: http://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml

[10] 4UrEyezOnly – A WP7 Exclusive [online]. 2011 [cit. 2013-05-07]. Dostupné z: http://www.4ureyezonly.com/

[11] Wiz Messenger [online]. 2011 [cit. 2013-05-07]. Dostupné z: http://www.wiz.co/

[12] Kryptos [online]. 2011 [cit. 2013-05-07]. Dostupné z: http://www.kryptoscommunications.com/

[13] WhatsApp Messenger. Aplikace pro Android ve službě Google Play [online]. 2012, 1. červenec 2012 [cit. 2013-05-07]. Dostupné z: http://play.google.com/store/apps/details?id=com.whatsapp

[14] WhatsApp [online]. 2012 [cit. 2013-05-07]. Dostupné z: http://www.whatsapp.com/

[15] THORNTON, James. WhatsApp updates iOS app, adds encrypted messaging. In: OnSoftware [online]. 2012-08-28 [cit. 2013-05-07]. Dostupné z: http://onsoftware.en.softonic.com/whatsapp-updates-ios-app-adds-encrypted-messaging

[16] CEJAS, Esteban. Download WhatsApp SNIFFER (APK) – READ WhatsApp CONVERSATIONS FROM OTHER PEOPLE. In: Android ADN: Android news apps and games [online]. 2011 [cit. 2012-12-08]. Dostupné z: http://androidadn.com/2012/06/download-whatsapp-sniffer-apk-read-whatsapp-conversations-from-other-people/

[17] Secret Message Elite. Aplikace pro Android ve službě Google Play [online]. 2012, 1. červenec 2012 [cit. 2013-05-07]. Dostupné z: http://play.google.com/store/apps/details?id=com.kalilabs.android.secretMessageElite

[18] BORISOV, Nikita, Ian GOLDBERG a Eric BREWER. Off-the-Record Communication, or, Why Not To Use PGP [PDF]. 2004 [cit. 2013-05-07]. Dostupné z: http://www.cypherpunks.ca/otr/otr-wpes.pdf

[19] Off-the-Record Messaging [online]. [2004] [cit. 2012-12-08]. Dostupné z: http://www.cypherpunks.ca/otr/

[20] RFC 6101. The Secure Sockets Layer (SSL) Protocol Version 3.0. Fremont (California): The Internet Engineering Task Force, 2011. Dostupné z: http://tools.ietf.org/html/rfc6101

[21] Features – GnuPG.org. The GNU Privacy Guard – GnuPG.org [online]. 2002-2004 [cit. 2013-05-07]. Dostupné z: http://www.gnupg.org/features.en.html

[22] RENDON, Casey. ComScore: Android remains on top in the US, but iOS makes gains. ComScore: Android remains on top in the US, but iOS makes gains [online]. 2013 [cit. 2013-05-03]. Dostupné z: http://www.androidcentral.com/comscore-android-remains-top

[23] MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.

[24] Java SE Open-Source JDK. ORACLE CORPORATION. Oracle: Hardware and Software, Engineered to Work Together [online]. 2013 [cit. 2013-05-05]. Dostupné z: http://www.oracle.com/technetwork/java/javase/community/opensourcejdk-jsp-136417.html

[25] Plausible Deniability. T r u e C r y p t: Free open-source disk encryption software for Windows 7/Vista/XP, Mac OS X, and Linux [online]. 2013, February 19, 2013 [cit. 2013-05-03]. Dostupné z: http://www.truecrypt.org/docs/?s=plausible-deniability

[26] YANG, Herong. SSL Client Authentication. In: JDK (Java Development Kit) Tutorials [online]. 2012 [cit. 2013-05-10]. Dostupné z: http://www.herongyang.com/JDK/SSL-Client-Authentication.html

[27] Java SE at a Glance. In: Oracle: Hardware and Software, Engineered to Work Together [online]. [2012] [cit. 2013-05-05]. Dostupné z: http://www.oracle.com/technetwork/java/javase/overview/index.html

[28] SQLite Home Page [online]. [2012] [cit. 2013-05-09]. Dostupné z: http://www.sqlite.org/

[29] About SQLite. SQLite Home Page [online]. 2013 [cit. 2013-05-05]. Dostupné z: http://www.sqlite.org/about.html

[30] SQL Features That SQLite Does Not Implement. SQLite Home Page [online]. 2013 [cit. 2013-05-05]. Dostupné z: http://www.sqlite.org/omitted.html

[31] Datatypes In SQLite Version 3. SQLite Home Page [online]. 2013 [cit. 2013-05-05]. Dostupné z: http://www.sqlite.org/datatype3.html

[32] Overview of MySQL Storage Engine Architecture. MySQL 5.5 Reference Manual [online]. [2010] [cit. 2013-05-07]. Dostupné z: http://dev.mysql.com/doc/refman/5.5/en/pluggable-storage-overview.html

[33] MEDNIEKS, Zigurd, Laird DORNIN, G. Blake MEIKE a Masumi NAKAMURA. Programming Android: Java Programming for the New Generation of Mobile Devices. 1st ed. Sebastopol: O'Reilly, 2011, xvi, 482 s. ISBN 978-1-449-38969-7.

[34] SQLite Foreign Key Support. SQLite Home Page [online]. 2013 [cit. 2013-05-07]. Dostupné z: http://www.sqlite.org/foreignkeys.html

[35] KLÍMA, Vlastimil. Nalézání kolizí MD5 na notebooku pomocí mnohonásobných modifikací zprávy. In: Nalézání kolizí MD5 na notebooku pomocí mnohonásobných modifikací zprávy [online]. verze 1, 2005 [cit. 2013-05-06]. Dostupné z: http://cryptography.hyperlink.cz/md5/Vlastimil_Klima_MD5_kolize.pdf

[36] BURR, William E. NIST Comments on Cryptanalytic Attacks on SHA-1. NIST.gov: Computer Security Division [online]. 2006, April 25, 2013 [cit. 2013-05-06]. Dostupné z: http://csrc.nist.gov/groups/ST/hash/statement.html

[37] Descriptions of SHA-256, SHA-384, and SHA-512. In: NIST.gov: Computer Security Division [online]. 2001 [cit. 2013-05-06]. Dostupné z: http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf

[38] LocationManager. Android Developes [online]. 2013, 01 May 2013 [cit. 2013-05-06]. Dostupné z: http://developer.android.com/reference/android/location/LocationManager.html

[39] HASHIMI, Sayed Y. Pro Android 2. New York: Apress, c2010, xvi, 718 s. Books for profesionals by professionals. ISBN 978-1-4302-2659-8.

[40] Google Maps Android API v2. Google Developers [online]. [2012], March 12, 2013 [cit. 2013-05-06]. Dostupné z: https://developers.google.com/maps/documentation/android/start

[41] Setup Google Play Services SDK. Android Developers [online]. [2011], [cit. 2013-05-06]. Dostupné z: http://developer.android.com/google/play-services/setup.html

[42] MEIER, Reto. Professional Android 2 application development. Indianapolis: Wiley, c2010, xxxii, 543 s. Wrox programmer to programmer. ISBN 978-0-470-56552-0.

Příloha A

Obsah CD

Přiložené CD obsahuje následující materiály:

Technická zpráva
: Soubor `technicka-zprava.pdf`

Zdrojové kódy technické zprávy
: Adresář `./zprava-tex`

Zdrojové kódy serverové části
: Adresář `./src-server`

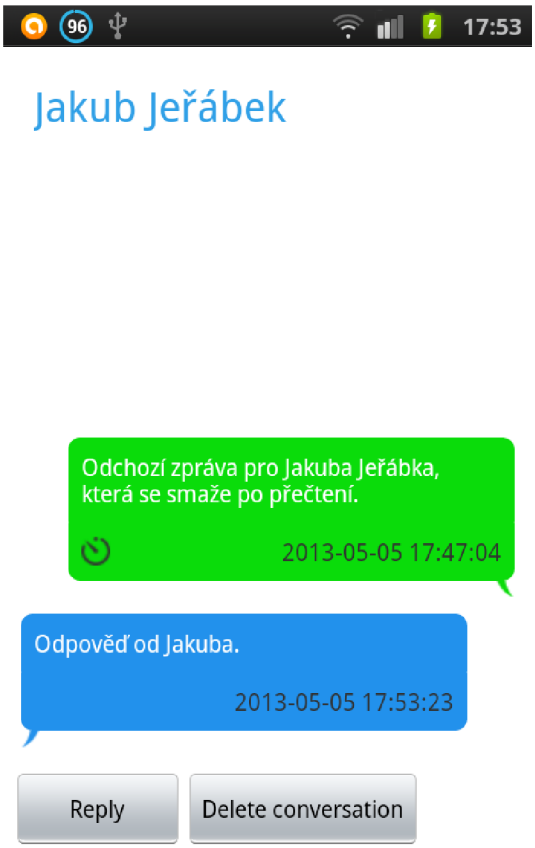
Zdrojové kódy klientské části
: Adresář `./src-client`

Pokyny ke zprovoznění aplikace
: Soubor `README.txt`

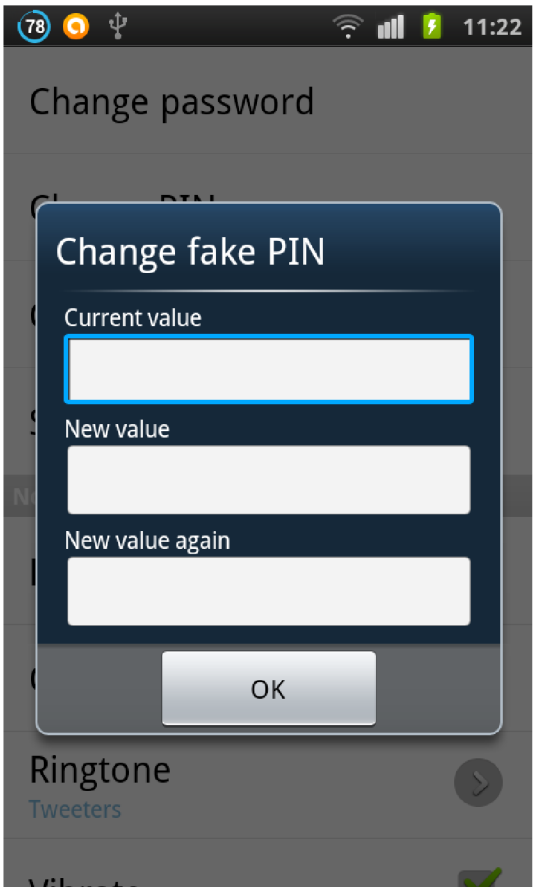
Příloha B

Snímky aplikace

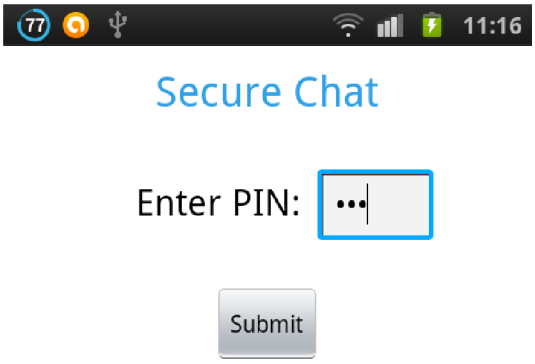
Na následujících obrázcích jsou k vidění snímky vybraných částí klientské aplikace.



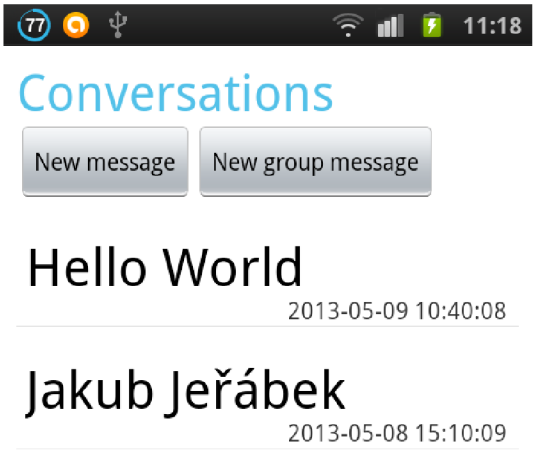
Obrázek B.1: Konverzace s uživatelem Jakub Jeřábek



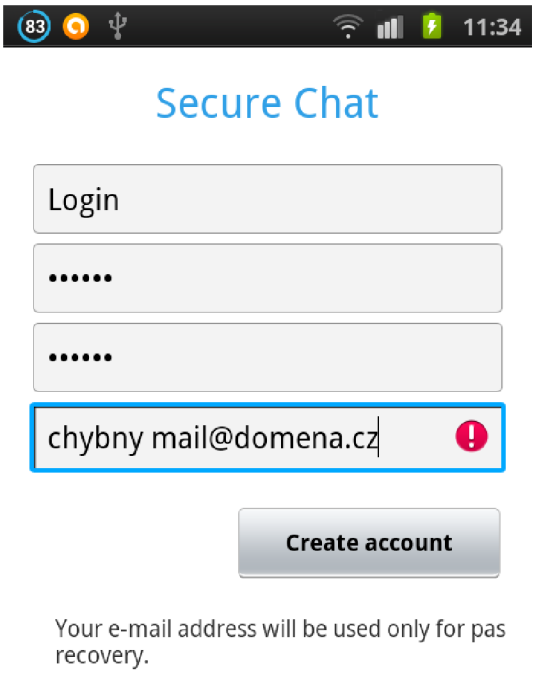
Obrázek B.2: Dialog v nastavení pro změnu falešného PIN kódu



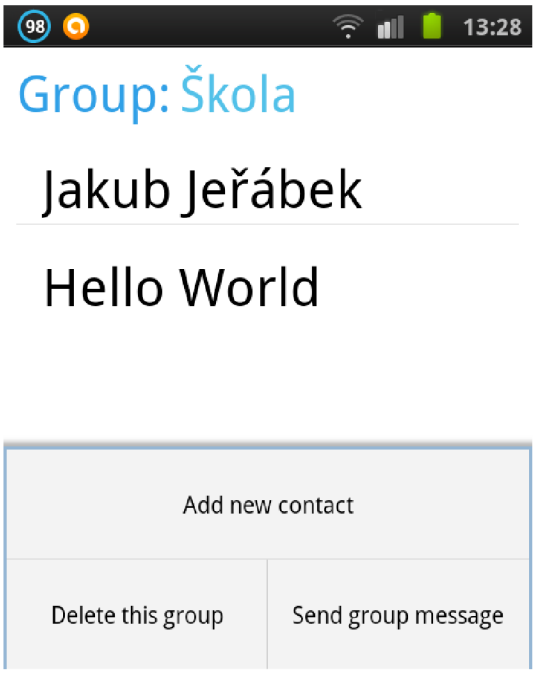
Obrázek B.3: Úvodní obrazovka vyžadující PIN kód



Obrázek B.4: Seznam konverzací s časem poslední zprávy



Obrázek B.5: Registrační formulář upozorňující na chybný e-mail



Obrázek B.6: Kontakty uložené ve skupině „Škola“