

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

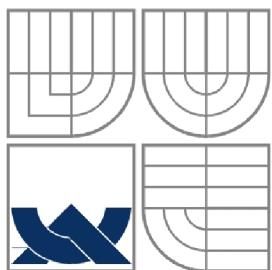
SPRÁVCE TEXTOVÝCH ZPRÁV PRO ANDROID

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

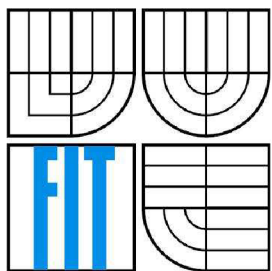
AUTOR PRÁCE
AUTHOR

JAN BLOUDÍČEK

BRNO 2012



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

SPRÁVCE TEXTOVÝCH ZPRÁV PRO ANDROID

TEXT MESSAGES MANAGER FOR ANDROID

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN BLOUDÍČEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. JAN KOUŘIL

Abstrakt

Tato práce se zabývá tvorbou aplikace pro mobilní zařízení, která slouží pro správu krátkých textových zpráv a elektronické pošty na platformě Android. Textové zprávy dovoluje odesílat také přes internetové brány. V práci jsou vysvětleny základní pojmy a technologie pro vývoj aplikací pro systém Android. Je zde popsána fáze analýzy, návrhu architektury a uživatelského rozhraní, implementace a testování programu.

Abstract

This thesis describes a creation of an application for mobile devices which is designed to manage short text messages and electronic mail on the Android platform. Text messages can be sent also through the SMS gateways. This work explains basic concepts and technologies for developing applications for Android. It describes the analysis phase, design of architecture and user interface, implementation and testing of the program.

Klíčová slova

Android, SMS, email, mobilní zařízení, SMS brána, uživatelské rozhraní, JavaScript

Keywords

Android, SMS, email, mobile device, SMS gateway, user interface, JavaScript

Citace

Jan Bloudíček: Správce textových zpráv pro Android, bakalářská práce, Brno, FIT VUT v Brně, 2012

Správce textových zpráv pro Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Kouřila. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Bloudíček
16.5.2012

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Janu Kouřilovi za odborné rady a čas při konzultacích.

© Jan Bloudíček, 2012.

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

Obsah

1 Úvod.....	7
2 Použité technologie.....	8
2.1 Historie Androidu.....	8
2.2 Architektura Androidu.....	8
2.3 Vývojové nástroje.....	9
2.3.1 Vývojové prostředí.....	10
2.3.2 Další nástroje.....	10
2.3.3 Zhodnocení možností vývoje.....	11
2.4 JavaScript na platformě Android.....	11
2.5 Síťové aplikace.....	11
2.6 Uživatelské rozhraní.....	12
2.6.1 Hlavní části.....	12
2.7 Publikace aplikací.....	13
3 Služba SMS a elektronická pošta.....	15
3.1 Služba SMS.....	15
3.1.1 Systém mobilní sítě s podporou SMS.....	15
3.1.2 Délka zprávy	16
3.1.3 Internetové brány.....	16
3.2 Elektronická pošta.....	17
4 Návrh Aplikace.....	18
4.1 Cílová skupina uživatelů.....	18
4.2 Definice funkcí aplikace.....	18
4.3 Průzkum trhu.....	19
4.4 Návrh uživatelského rozhraní.....	20
4.4.1 Tvorba prototypu.....	20
4.4.2 Popis obrazovek.....	21
4.5 Návrh tříd aplikace.....	21
4.5.1 Návrh databáze pro uložení elektronické pošty.....	23
4.6 Přidání nových internetových bran.....	23
4.7 Výběr verze systému.....	24
5 Implementace.....	25
5.1 Optimalizace aplikace.....	25
5.2 Odesílání zpráv přes internetové brány.....	25
5.2.1 Určení internetové brány podle čísla.....	26
5.2.2 Skripty internetových bran.....	27
5.3 Vytvoření grafického uživatelského rozhraní.....	27
5.3.1 Ovládání pohybem prstu.....	27
6 Testování aplikace.....	29
6.1 Úkoly k vypracování.....	29
6.1.1 Vytvoření zprávy.....	29
6.1.2 Zobrazení historie zpráv.....	30
6.1.3 Manipulace se zprávami.....	30
6.1.4 Dodatečné otázky.....	30
6.2 Výsledky testování.....	30
6.2.1 Test pro vytvoření nové zprávy.....	31
6.2.2 Test pro zobrazení historie zpráv.....	31
6.2.3 Testování manipulace se zprávami.....	31
7 Závěr.....	32
Literatura.....	33

Seznam příloh.....	34
Příloha 1: Obrazovky aplikace.....	35

1 Úvod

Tématem této bakalářské práce je vytvoření aplikace pro správu textových zpráv na platformě Android. Mobilní zařízení s pokročilými operačními systémy jsou v současné době velmi rozšířené. S tím souvisí i značný rozvoj aplikací pro tato zařízení. Jednou z hlavních funkcí nadále zůstává komunikace s okolím. Existuje řada způsobů, jak lze být ve spojení s využitím mobilního zařízení. Nejpoužívanější jsou telefonní hovory, textové a multimediální zprávy, elektronická pošta a odesílání a přijímání zpráv v reálném čase. Nabízí se možnost vytvoření programu, který sjednocuje některé ze způsobů komunikace do jediné aplikace. Tato práce se zabývá možností kombinace služeb SMS a elektronické pošty. Dále se snaží poskytnout funkce, které toho využívají a dávají uživateli více možností a větší komfort. Krátké textové zprávy byly a stále ještě jsou značně populární. Je to hlavně díky jejich dostupnosti, jelikož většina populace dnes vlastní mobilní telefon se službou SMS. Tyto zprávy mohou být odeslány normální placenou cestou, tedy přímo operátorovi mobilní sítě. Další možností je využití služeb poskytovatele, který je k operátorovi připojen. S příchodem chytrých telefonů se rozšířila tato funkce i na mobilní zařízení. Podobná situace je s elektronickou poštou. Ta je sice dostupná na modelech mobilních telefonů bez pokročilého operačního systému, ale velkého úspěchu se zde nedočkala.

Cílem této práce je vytvořit aplikaci, která bude umožňovat odesílání krátkých textových zpráv přes internetové brány všech českých operátorů a také elektronické pošty. Dále je předmětem vývoje možnost rozšíření funkcí programu o postupy pro odesílání zpráv přes internetové brány, které nejsou jeho součástí. Důležitým aspektem je navrhnout program s ohledem na změnu protokolu na straně brány a zaměřit se na uživatelskou přívětivost.

2 Použité technologie

Moderní mobilní zařízení, jako například chytré telefony a tablety, často využívají pokročilé operační systémy. Jejich výhodou je, že jsou obvykle nezávislé na technickém vybavení zařízení, na kterém jsou spuštěny. Další výhodnou vlastností je vývoj nových aplikací, které rozšiřují možnosti zařízení. Jako nejúspěšnější lze zmínit iOS, Symbian, Windows Mobile, BlackBerry a Android.

Android je platforma pro mobilní zařízení, což zahrnuje například telefony, malé kapesní počítače (Personal Digital Assistant), tablety, netbooky, navigace pro automobily a další. Obsahuje nejen operační systém, ale i mezivrstvu poskytující rozhraní pro vytváření aplikací a základní sadu aplikací. Android je šířen jako otevřený software, což zvyšuje jeho atraktivitu pro nasazení do nejrůznějších zařízení.

2.1 Historie Androidu

U zrodu platformy Android stála do té doby nepříliš známá společnost Android Inc. Od začátku bylo hlavním záměrem vytvořit nový systém pro mobilní zařízení. Tuto firmu odkoupila v roce 2005 společnost Google i s projektem Android. V nadcházejícím období probíhal rozvoj tohoto systému [11]. Dalším důležitým milníkem je vznik Open Handset Alliance v roce 2007. Je to mezinárodní aliance významných firem v oblasti informatiky a komunikačních technologií. Hlavním cílem tohoto uskupení bylo vytvořit platformu pro mobilní zařízení, která by byla otevřená. Právě otevřenost je jednou z cenných vlastností pro firmy i programátory [12].

První verze systému, která byla dostupná veřejnosti na zařízení T-Mobile G1, byla vydána v únoru roku 2009. Pak následoval rychlý vývoj a postupně byly přidávány klíčové vlastnosti. Mezi ně lze zařadit podporu softwarové klávesnice a Bluetooth ve verzi Android Cupcake, dále pak postupně vylepšení uživatelského rozhraní. Také sem patří NFC¹ a možnost instalace aplikací na externí úložiště ve verzi Android Froyo a další. Byly vyvinuty verze určené primárně pro tablety, které mají označení Android Honeycomb. Do té doby tato zařízení využívala systém navržený pro mobilní telefony. Nejnovější verze systému je Android Ice Cream Sandwich, jejíž hlavní výhodou je, že je určená pro většinu typů zařízení běžících na platformě Android. Jedná se především o tablety a mobilní telefony [18].

2.2 Architektura Androidu

Jak už bylo zmíněno, Android je složen z operačního systému a mezivrstvy, která poskytuje rozhraní pro vytváření aplikací. Hlavními součástmi operačního systému jsou jádro operačního systému, prostředí pro běh aplikací, knihovny, služby systému a základní sada aplikací [14].

- Jádro operačního systému je založené na operačním systému Linux verze 2.6. Poskytuje základní služby pro správu paměti a procesů. Dále je zde vrstva, která slouží jako rozhraní mezi fyzickým zařízením a zbytkem programového vybavení.
- Prostředí pro běh obsahuje některé hlavní knihovny z programovacího jazyka Java. Důležitým prvkem je virtuální stroj² Dalvik. Na tomto stroji jsou spuštěny programy ve formátu dex. Ten nabízí oproti Java bytekódu³ optimalizace za účelem snížení nároků na zdroje, které jsou kvůli omezením mobilních zařízení kritické. Jedná se především o paměť a výkon procesoru.

1 NFC (Near Field Communication) – technologie pro bezdrátovou komunikaci zařízení na krátkou vzdálenost.

2 Virtuální stroj - software, který poskytuje prostředí pro běh aplikací.

3 Bytekód (Bytecode) – reprezentace mezikódu v binární formě, která je platformě nezávislá.

Android je koncipován tak, že každá aplikace je spuštěna ve vlastním procesu se svojí instancí stroje Dalvik.

- Další důležitou částí systému jsou knihovny, které nabízejí podporu pro vývoj aplikací. Jsou zde obsaženy například knihovny pro práci s databází, médii, počítačovou grafikou a další.
- Služby systému pak rozšiřují knihovny uvedené v minulém odstavci a poskytují je vývojářům. Jsou zde obsaženy například služby pro správu elementárních částí aplikace a poskytování dat napříč programy.
- Android nabízí i sadu aplikací pro zajištění běžných funkcí zařízení. Mezi ně můžeme zařadit emailového klienta, prohlížeč internetových stránek a kalendář.

2.3 Vývojové nástroje

Android nabízí velké množství nástrojů pro podporu a urychlení vytváření aplikací. V tomto oddílu budou popsány ty, které byly použity.

Při vývoji aplikací pro platformu Android máme na výběr z několika přístupů. Každý z nich má své výhody, nevýhody a oblasti použití. Výběr závisí především na požadavcích, které klademe na výslednou aplikaci, její nároky na výkon, komplexnost a další kritéria. Máme tedy na výběr z těchto možností:

- Android SDK,
- Android NDK,
- HTML,
- interpretované programovací jazyky.

Standardním nástrojem pro vývoj aplikací je Android SDK (Software Development Kit). Dává nám k dispozici velké množství knihoven, nástrojů a služeb, které vývoj do značné míry usnadňují. Dodatečně si můžeme nainstalovat software pro testování aplikací. Ten nám dovoluje vytvořit virtuální zařízení, které slouží jako předpis pro emulátor. S jeho pomocí otestujeme aplikaci jako na skutečném zařízení. Náhled emulátoru lze nalézt na obrázku 2.1. Programovacím jazykem je zde Java, která vychází z Java SE. S ní však není plně kompatibilní. Například neobsahuje knihovnu Swing pro práci s grafickým uživatelským rozhraním, ale definuje vlastní. Android SDK je vhodný pro většinu aplikací, které nemají příliš velké požadavky na výkon [15].

Android NDK (Native Development Kit) je přídatný software k SDK a využívá se především v programech, které mají velké nároky na výkon. Jazykem pro vývoj je v tomto případě C++. Z toho plyne i další použití NDK, a to využití kódu, který je již vytvořený v jazyce C++ nebo C. Není tedy nutné takovýto program přepisovat do jazyka Java. Je zde však i několik nevýhod. Zejména



Obrázek 2.1: Emulátor pro platformu Android.

je to snížení kompatibility mezi různými verzemi systému Android a také nemáme k dispozici kompletní rozhraní pro programování jako v SDK. Navíc je kód složitější z hlediska správy i testování [16].

Aplikace lze dokonce vyvíjet za pomoci HTML, což může být pro mnohé programátory lákavé. Ovšem značnou nevýhodou je, že uživatelské rozhraní je tvořeno ovládacími prvky tohoto značkovacího jazyka. Pokud bychom se rozhodli použít pouze HTML, pak by realizace některých částí programu mohla být náročnější. Prvky rozhraní systému Android jsou totiž lépe optimalizovány pro mobilní zařízení.

Další možností je použít interpretované programovací jazyky. V programu máme k dispozici většinu knihoven a služeb systému Android a lze tak vytvářet plnohodnotné aplikace. Tento způsob slouží zvláště pro psaní jednodušších programů, jejichž vývoj může být v porovnání se standardním přístupem rychlejší. Navíc lze integrovat knihovny interpretu do naší aplikace, která je napsaná například v Javě a vykonávat skripty tam. Android však v sobě nezahrnuje žádný interpret těchto jazyků a musíme tedy pro ně přidat podporu dodatečně. Za tímto účelem vzniklo několik projektů, z nichž asi nejrozsáhlejší je SL4A (Scripting Layer for Android). Nabízí vše potřebné a v současnosti poskytuje interprety pro jazyky Python, Perl, JRuby, Lua, BeanShell, JavaScript a Tcl [17]. Velmi zajímavý je BeanShell. Jedná se o interpretovaný jazyk, který je podobný Javě. Je to asi nejsnazší volba, jak provádět skripty na systému Android. Dalším jazykem, který mě zaujal, je JavaScript. Tento jazyk je velmi populární a známý. Programování v něm by tedy nemělo činit problémy nikomu, kdo se vývojem aplikací alespoň trochu zabýval.

2.3.1 Vývojové prostředí

Když už víme, jaký přístup zvolit pro implementaci naší aplikace, můžeme si nainstalovat prostředí pro její tvorbu. Výchozí prostředí pro Android je Eclipse IDE, které dovoluje vyvíjet aplikace s pomocí Android SDK a také NDK. Pro integraci nástrojů potřebných k vývoji slouží zásuvný modul ADT⁴. Samozřejmě jsou podporována i jiná prostředí, jako například NetBeans. Nebo lze psát aplikaci pouze s využitím Android SDK, nějakého textového editoru a nástroje Ant známého programátorům v jazyce Java. Ten slouží pro automatizované překládání, spouštění a testování aplikací.

2.3.2 Další nástroje

Pomůcek pro usnadnění tvorby programu pro Android je opravdu mnoho a popis všech je nad rámec tohoto dokumentu. Rozhodně bychom však neměli opomenout DDMS (Dalvik Debug Monitor Server). Tento nástroj slouží především pro ladění a optimalizaci programu. Využívá se i k simulaci některých služeb, které jsou běžně dostupné na mobilních zařízeních. Patří sem příchozí hovory, krátké textové zprávy a poloha zařízení. Pro pokročilou analýzu programu ve směru správy paměti je neocenitelným nástrojem MAT (Memory Analyzer Tool). Používá se ve spojení s DDMS a poskytuje hodnotné informace při optimalizaci aplikace.

Dalším užitečným pomocníkem je Layout Editor. Je to grafický editor pro vytváření a interaktivní změnu uživatelského rozhraní, který dokáže vygenerovat kód ve formě XML. Avšak vytváření složitějšího rozhraní je v něm obtížnější, než ho napsat ručně. Je ale prospěšný, pokud chceme výsledek návrhu zobrazit. Jinak bychom museli program přeložit, sestavit a nahrát do emulátoru nebo zařízení, což nějakou chvíli trvá.

Optimalizace struktury uživatelského rozhraní je stejně důležitá jako u zbytku kódu. Platí zde několik pravidel, které bychom měli dodržovat, pokud chceme dosáhnout efektivního programu a současně kompatibility s nadcházejícími verzemi systému Android. Důležité je příliš nezanořovat prvky do sebe. Jakmile dosáhneme úrovně, která přesahuje 10 vrstev, měli bychom uvažovat o lepším návrhu. Nástroj, který lze v tomto případě využít, je Hierarchy Viewer. Poskytuje nám přehledné zobrazení struktury obrazovky, díky kterému můžeme návrh zlepšit.

⁴ ADT (Android Developer Tools) - zásuvný modul pro vývojové prostředí Eclipse, který poskytuje nástroje pro jednodušší vývoj aplikací na platformě Android.

2.3.3 Zhodnocení možností vývoje

Možností pro vývoj programů pro platformu Android je několik. V předchozích odstavcích byly shrnuty jednotlivé přístupy a popsány jejich výhody a nevýhody. Zvolení vhodného nástroje pro vývoj aplikace bylo dáno jejími nároky na výkon a také poměrně velkou rozsáhlostí. Použití nástroje Android NDK je nevýhodné, protože by neměl pro tento program žádný přínos. Zvolil jsem tedy jen prostředí SDK, které nabízí řadu knihoven a nástrojů pro jednodušší a rychlejší vývoj aplikací. Dále je v aplikaci zabudovaný interpret jazyka JavaScript kvůli možnosti přidávat postupy pro odesílání zpráv na internetové brány, které v ní nejsou implementovány. Na výběr bylo více jazyků vhodných pro tento účel. Jako příklad lze uvést BeanShell. JavaScript byl zvolen zejména díky své velké oblíbenosti mezi programátory. K nim totiž patří i vývojáři internetových stránek, kteří tuto skupinu značně rozšiřují.

2.4 JavaScript na platformě Android

Programovací jazyk JavaScript je známý asi každému, kdo se programováním alespoň trochu zabýval. Interpret tohoto jazyka přímo pro Android nenajdeme. Existuje ale pro jazyk Java, který můžeme s několika úpravami využít. Projekt Rhino je implementací jazyka JavaScript a je napsán v jazyce Java. Byl vyvinut firmou Netscape jako součást jejího prohlížeče Javagator. Jednalo se o verzi prohlížeče Navigator, který měl být kompletně přepsán do jazyka Java. Projekt však nebyl příliš úspěšný, ale zbylo po něm právě Rhino s potenciálem dalšího využití. Rhino je volně šiřitelný software a v současné době je spravován organizací Mozilla [7].

Projekt umožňuje využití skriptů napsaných v jazyce JavaScript uvnitř aplikací, které jsou napsány v Javě. Skripty mohou být prováděny buď v kompilovaném, nebo v interaktivním režimu. V kompilovaném režimu je přeložen zdrojový kód do bytekódu, který je dále vstupem virtuálního stroje. V interaktivním režimu je kód skriptu proveden přímo interpretem. Kdybychom chtěli skript spouštět opakovaně, mohlo by být použití kompilovaného režimu efektivnější.

Pokud chceme přenést Rhino na systém Android, musíme počítat s několika obtížemi. V kompilovaném módu je využit JIT⁵ kompilátor, který pracuje s bytekódem Javy. Ten však není kompatibilní s formátem bytekódu, který používá Android. Kompilovaný mód tedy na Androidu využít nelze. Můžeme ale použít interpret [6]. Dále velikost knihovny není zanedbatelná, v době tvorby programu měla přibližně 600 kB. Měli bychom proto uvážit, zda využijeme ve skriptech všechny její funkce a případně některé z knihovny odebrat.

Rhino je v tomto projektu využito pro přidávání postupů pro odesílání krátkých textových zpráv na internetové brány. Pro naše potřeby není kompilovaný režim ani nutný, plně vyhovuje interpret. Poslední verze Rhina podporuje JavaScript verze 1.7 a tato verze je také přítomna v aplikaci. Z ní byly odebrány některé funkce, jako E4X, JavaAdapter, optimalizátor pro kompilovaný režim a některé další nástroje, které nejsou ve skriptech využity. Velikost výsledného souboru Rhina je tak výrazně menší. Celý postup redukce funkcí knihovny a s tím spojené zmenšení velikosti výsledného souboru je popsán na internetových stránkách Rhina.

2.5 Síťové aplikace

Komunikace s okolním světem je pro mobilní zařízení základní vlastností a lze ji realizovat mnoha způsoby. Může se jednat o připojení k internetu prostřednictvím datových služeb, jako GRPS, EDGE, 3G nebo síť Wi-Fi. Dále je k dispozici připojení k síti GSM nebo CDMA a jistě byste našli i další. Tato kapitola je věnována možnostem, které jsou při tvorbě programu použity.

5 JIT (Just In Time) - metoda překladu zdrojového kódu pro zefektivnění běhu programu.

Systém Android nabízí při programování síťových aplikací mnoho možností. Lze využít různé úrovně přístupu vzhledem k tomu, co požadujeme. Na té nejvyšší úrovni jsou nástroje zabudované v Androidu, které poskytují zapouzdření pro veškeré potřebné funkce pro připojení k síti. Například jádro WebKit, což je nástroj pro tvorbu prohlížeče internetových stránek. Pro většinu protokolů jsou již vytvořeny knihovny. Jedná se často o projekty třetích stran, které nejsou součástí Androidu. Jako příklad můžeme zmínit knihovnu JavaMail, respektive její verzi pro Android. Poskytuje funkce pro přijímání a odesílání emailů, autentizaci, metody pro bezpečný přenos dat po síti a další. Tato knihovna je využita v tomto projektu, a to hlavně pro odesílání elektronické pošty pomocí protokolu SMTP, autentizaci a zabezpečený přenos dat. Další protokol, který byl použit je HTTP a HTTPS. Podpora pro komunikaci pomocí HTTP protokolu je v systému vestavěná. Dále jsou zde nástroje pro práci s nižšími vrstvami ISO/OSI modelu, jako sokety a surové sokety.

2.6 Uživatelské rozhraní

Uživatelské rozhraní je spojnicí mezi člověkem a strojem nebo systémem. Pomáhá mu tyto nástroje ovládat a pracovat s nimi. V prostředí počítačů se rozhraní obvykle skládá z nějakého zobrazovacího zařízení a dále prvků pro ovládání, jako je například klávesnice. Můžeme se setkat s různými typy uživatelských rozhraní. V dalším textu bude bráno v úvahu pouze grafické uživatelské rozhraní (Graphical User Interface), které je na mobilních zařízeních nejčastější. Typicky se skládá z několika částí, jako například dialogy, okna a řídicí prvky pro ovládání systému [10].

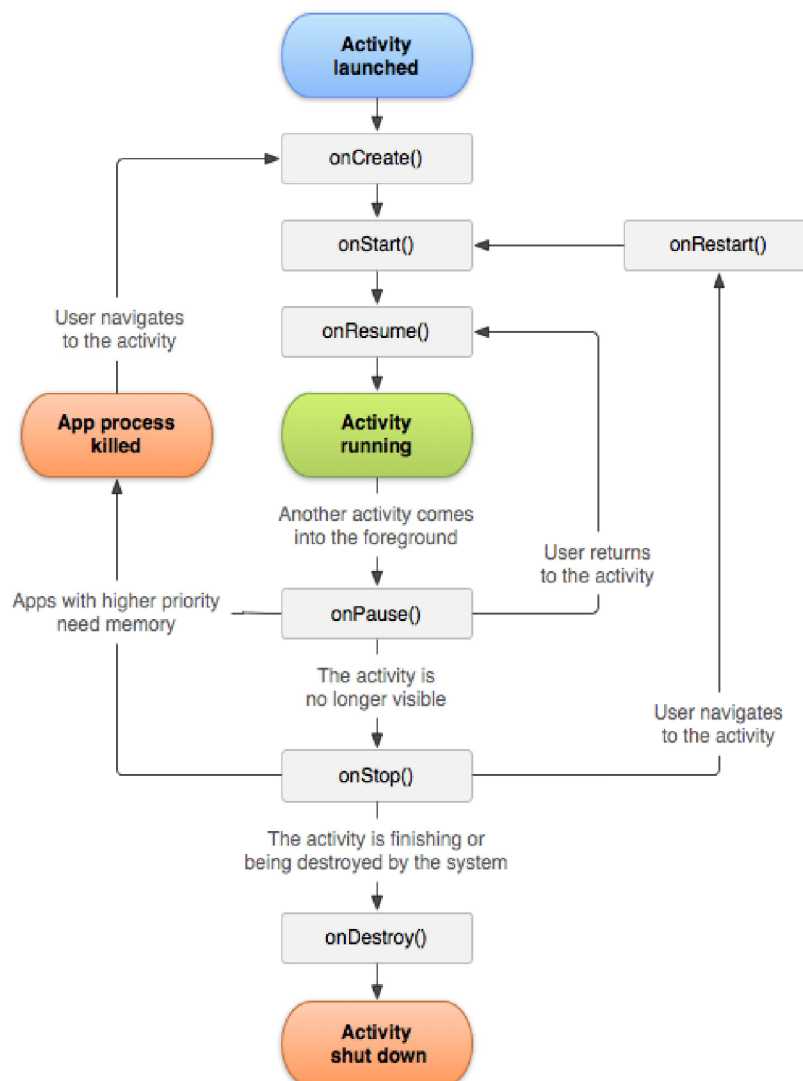
2.6.1 Hlavní části

Uživatelské rozhraní systému Android tvoří několik základních elementů. Pochopení jejich významu a oblasti použití je nutnou podmínkou pro vytvoření kvalitní aplikace. Jsou to především prvky:

- Aktivita (Activity),
- pohled (View),
- kontejner (ViewGroup).

Aktivita (Activity) je jednou ze základních komponent uživatelského rozhraní a představuje právě jednu obrazovku. Prochází životním cyklem, který definuje všechny stavy, ve kterých se může nacházet. Přejít mezi stavy je reakcí na podněty uživatele a je mu přidělena alespoň jedna metoda. Díky ní může naše aplikace na tyto události reagovat. Jako příklad lze uvést přepnutí na jinou obrazovku, ukončení stávající nebo přechod zařízení do režimu spánku. Celý proces je znázorněn na obrázku 2.2. Všechny vytvořené aktivity jsou umístěny v zásobníku aktivit. Na jeho vrcholu může být vždy jen jedna, a to ta, která je aktuálně viditelná. Nikdy tak nemůže nastat případ, že by se zobrazila obrazovka předešlá a nová by nebyla ukončena [9]. Jelikož obrazovka mobilních zařízení nenabízí příliš mnoho prostoru, je nutné rozložit činnosti do více aktivit. Každá obrazovka by měla mít jeden jednoduchý účel. Musí tedy existovat způsob, jak spustit novou aktivitu z té stávající. K tomuto nám Android poskytuje záměry (Intent). Objekt, který představuje záměr, určíme, co chceme udělat a s čím. Dále jen necháme systém, aby podle něj vyhledal vhodnou komponentu a spustil ji. Může se jednat o aktivitu, službu, jinou aplikaci a jistě byste našli i další cíle.

Obsah obrazovky tvoří prvky, které se nazývají pohledy. Jako pohled lze označit vše, co vidíme na obrazovce. Patří sem ovládací prvky, jako tlačítka, popisky a vstupní textová pole. Můžeme si dokonce vytvořit vlastní ovládací prvky nebo si přizpůsobit stávající. K tomu by měl být vždy dobrý důvod, abychom zbytečně nemátli uživatele, který je zvyklý na již zavedený vzhled a způsob používání. Nové prvky lze vytvořit několika způsoby. Je možné využít knihovnu OpenGL pro tvorbu 2D a 3D grafiky. Dalším způsobem je práce s existujícími pohledy, jejich vhodná kombinace a upravení.



Obrázek 2.2: Životní cyklus aktivity (převzato z [9]).

Zvláštním typem pohledu je kontejner (ViewGroup). Jeho úkolem je seskupovat jednotlivé pohledy, definovat jejich polohu a vzájemné vztahy na obrazovce. Patří sem například LinearLayout, RelativeLayout a TableLayout. Způsob tvorby

Uživatelské rozhraní je možné definovat dvěma způsoby. Může být součástí kódu napsaného v jazyce Java, což se používá zejména v případech, kdy dopředu nevíme, kolik nebo jaké prvky budou zobrazeny. Tento přístup se však nehodí pro běžný návrh. Daleko výhodnější je specifikovat rozhraní ve formátu XML. Definice zobrazované části aplikace je tak oddělena od jejího zbytku, což zlepšuje přehlednost programu. Navíc můžeme využít nástroje, které pracují s XML soubory a návrh nám usnadňují.

2.7 Publikace aplikací

Pokud již máme aplikaci funkční a vyzkoušenou na reálných zařízeních, můžeme přistoupit k jejímu zveřejnění. Nejvhodnějším místem je služba Google Play, která kromě aplikací nabízí i multimediální

obsah. Abychom zde mohli aplikaci publikovat, musíme se nejdříve zaregistrovat a zaplatit poplatek 25 dolarů. Ten mimo jiné slouží pro odfiltrování aplikací, které mají charakter spamu.

Minimální krok, který je nutné před uvolněním aplikace podstoupit, je podepsání balíčku našeho programu certifikátem. Společnost Google používá tento systém certifikátů ke spojení aplikace a jejího tvůrce, kterého lze takto jednoznačně identifikovat. Zároveň slouží jako důležitá bezpečnostní ochrana uživatele i programátora. Uživatel může celkem jednoduše zjistit, kdo aplikaci vytvořil. Vývojář má jistotu, že nikdo cizí nevydává pod jeho jménem například škodlivé nebo nekvalitní programy. Certifikát nemusí být podepsaný žádnou certifikační autoritou. Postačí, když bude podepsaný jeho tvůrcem. Systém pro nahrání aplikace na internet po nás také bude požadovat další dodatečné soubory. Jedná se především o propagační grafiku, která bude náš program reprezentovat. Další podrobnosti lze najít ve [13].

3 Služba SMS a elektronická pošta

Tato kapitola popisuje základní koncept a prvky služby SMS a systému elektronické pošty. Vytvářený program bude sloužit pro odesílání krátkých textových zpráv a emailů, a proto bylo nutné se těmito systémy zabývat. Služba SMS je zde popsána detailněji než elektronická pošta, neboť je pro vytvářený program klíčová.

3.1 Služba SMS

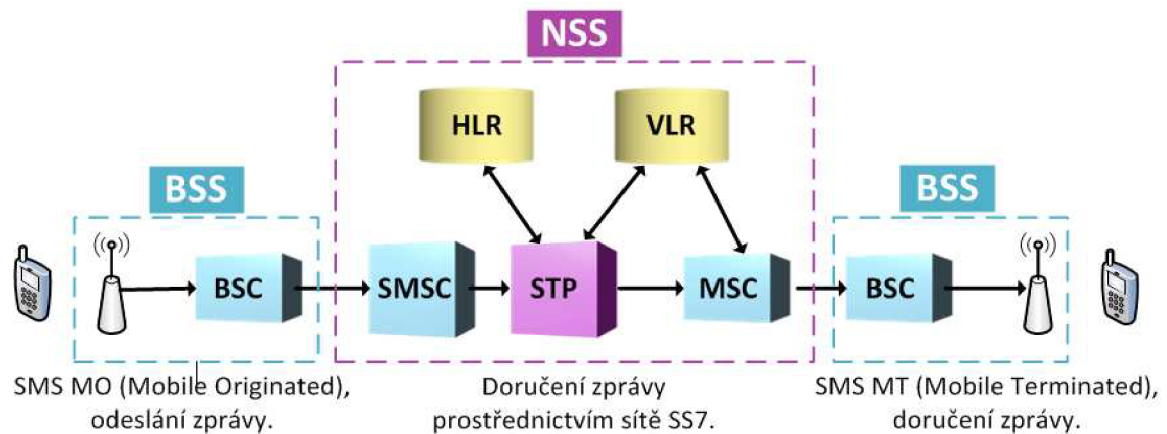
Technologie SMS (Short Message Service) má velký význam mezi prostředky pro komunikaci na mobilních zařízeních. Je to služba, která umožňuje přijímání a odesílání krátkých textových zpráv. SMS je jedna z nejrozšířenějších forem pro sdílení textu na mobilních zařízeních, i když se v poslední době rozšířily také jiné formy přenosu. Úspěch textových zpráv zaručuje několik věcí. Jednou z nich je její jednoduchost z hlediska použití a dostupnosti. Téměř každý dnes má mobilní telefon s touto službou. Když odešleme zprávu, tak víme, že s velkou pravděpodobností si ji příjemce přečte v krátkém časovém intervalu. Dalším faktorem zajišťujícím úspěch je závazek mobilních operátorů a přidružených firem, který byl uzavřen v roce 1995 a nese název GSM MoU Association. Mimo jiné zajišťuje, že každé mobilní zařízení a síť by měla podporovat SMS službu.

Začátky SMS služby sahají do období 1984-1985, kdy se stanovil počáteční koncept. Zanedlouho, v roce 1987 již byla na světě její první standardizovaná podoba, která byla součástí standardu GSM⁶. Vývoj však tímto nekončil. V následujících letech byl standard upravován, přidávány nové funkce a rozšíření [4].

3.1.1 Systém mobilní sítě s podporou SMS

Hlavní prvky systému jsou na obrázku 3.1. Jedním ze základních stavebních prvků mobilní telefonní sítě je mobilní stanice, pomocí které se můžeme připojit k systému BSS. BSS (Base Station Subsystem) zajišťuje radiové spojení pro mobilní zařízení a tvoří síť pokrývající oblast signálem. Dalším prvkem sítě je NSS (Network Switching subsystem), který plní funkci mobilní ústředny a poskytuje především spojovací funkce a funkce související s možností pohybu mobilních stanic [1]. Mezi nejdůležitější komponenty systému NSS patří MSC (Mobile Switching Centre), které zajišťuje zmíněné funkce pro spojení mobilních stanic se sítí a jejich mobilitu. Také jsou zde databáze pro uchování dat o účastnících, mobilních stanicích a další informace spojené s řízením hovorů. Prvek STP (Signaling Transfer Point) plní podobnou funkci jako směrovač v IP sítích a slouží například pro směrování zpráv. Nezbytnou součástí NSS pro poskytování SMS služeb je SMSC (SMS Center), jež umožňuje zpracování zpráv. Poskytuje mechanismus store-and-forward. Při přijetí zprávy nejdříve zjistí, zda je příjemce dostupný. Pokud ne, zprávu uloží a po nějaké době se ji pokusí doručit znovu. Není však zaručeno, že zpráva bude příjemci doručena. Jedná se o tzv. „best effort“ [2].

6 GSM (Global System for Mobile Communications) – standard pro celulární síť s digitálním přenosem.



Obrázek 3.1: Struktura mobilní sítě s podporou SMS.

3.1.2 Délka zprávy

Maximální délka zprávy je určena protokolem, kterým jsou data přenášena mezi MSC jednotkami. Používá se zde protokol MAP⁷ ze skupiny signalizačních protokolů SS7⁸. Maximální velikost zprávy je tedy 140 bytů [4]. Zpráva však může obsahovat až 160 znaků bez diakritiky. Je to dáno znakovou sadou definovanou v GSM 03.38, která na jeden znak přiřazuje 7 bitů. Při využití diakritiky se používá 16 bitové kódování znaků UCS-2. Pokud zpráva přesáhne maximální počet znaků, je rozdělena na části, jež jsou odeslány odděleně a na cílovém zařízení opět složeny do jediné zprávy [3]. Pokud pro odeslání využijeme internetovou bránu, pak musíme zjistit sami, jaká délka zprávy je podporována. Většinou se jedná o 160 znaků, ale nelze se na to spoléhat. V případě, že bychom odeslali zprávu delší, než jakou je brána ochotná přijmout, byla by zkrácena na vyhovující délku.

3.1.3 Internetové brány

SMS zprávu lze odeslat i přes poskytovatele, který nemusí být přímo operátorem. Je tedy možno využít internetové brány, které zprostředkovávají SMS služby. Internetová brána je systém, který je napojený na SMS centrum a směřuje k němu zprávy k dalšímu zpracování. Poskytuje rozhraní s využitím protokolu HTTP nebo HTTPS. Pomocí metod POST a GET lze bráně posílat požadavky a přijímat odpovědi. Metodou POST se předává zpráva k odeslání, číslo odesílatele a další data, která jsou vyžadována. Některé brány poskytují dodatečné funkce, jako potvrzení o přijetí zprávy, vyplnění čísla odesílatele a další. Odpověď může být soubor ve formátu JSON, XML nebo HTML. Především je v této odpovědi obsažena informace o tom, zda byla zpráva úspěšně předána ke zpracování nebo došlo k chybě. Pokud nastala nějaká chyba, je v odpovědi obvykle popsána její příčina.

7 MAP (Mobile Application Part) – aplikační protokol z rodiny protokolů SS7.

8 SS7 (Signalling System No. 7) – skupina řídicích a signalizačních protokolů pro telefonní služby.

3.2 Elektronická pošta

Mobilní zařízení s pokročilým operačním systémem, jako je například Android, mohou přistupovat k internetové síti více způsoby. Lze využít mobilní datové služby, například GPRS, EDGE. Dále je k dispozici Wi-Fi. Můžeme tedy velmi snadno a levně přistupovat k internetové síti a pracovat s elektronickou poštou. Elektronická pošta je způsob, jak přenášet digitální data po síti. Poskytuje možnost přenosu zpráv, jejichž obsah může být různého typu. Tato služba se skládá ze tří základních komponent:

- Klient – MUA (Mail User Agent),
- server – MTA (Mail Transfer Agents),
- protokol – SMTP (Simple Mail Transfer Protocol).

Klient je zastoupen programem na straně uživatele pro psaní a čtení zpráv a také pro manipulaci s emailovou schránkou. Server slouží pro přenesení zpráv mezi zdrojem a cílem. Jeho součástí může být MSA (Mail Submission Agent), který přebírá zprávy od klienta a předává je cílovému serveru a navíc poskytuje některé pokročilé funkce. SMTP je internetový protokol sloužící pro výměnu zpráv. Tento protokol se používá pro přenos dat od uživatele na server a také pro výměnu zpráv mezi servery.

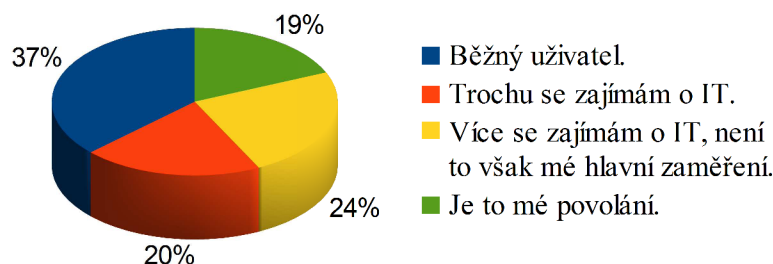
Email má definovanou strukturu a skládá se z obálky a zprávy. Obálka poskytuje zapouzdření pro položky protokolu. Obsahuje například adresu odesílatele a adresy příjemců, pokud jich je více. Druhá část emailu, tedy zpráva, má dvě části, a to hlavičku a tělo. V hlavičce můžeme nalézt například datum odeslání a také informace o odesílateli emailu. Tělo může obsahovat text, ale i binární data [5].

4 Návrh Aplikace

Tato kapitola popisuje návrh aplikace a uživatelského rozhraní. Fáze návrhu je velmi důležitou součástí vývoje a neměla by být zanedbána, i když se tak často děje. Pokud aplikaci správně navrhne, ušetříme si mnoho času při implementaci.

4.1 Cílová skupina uživatelů

Pro dobrý návrh aplikace je nezbytné vědět, jaká je cílová skupina uživatelů. Službu SMS a elektronickou poštu dnes používá téměř každý, nehledě na věkovou kategorii a povolání. Různorodost potencionálních uživatelů je tedy opravdu velká. Abych alespoň trochu upřesnil budoucí uživatele a zároveň získal další informace vhodné pro pozdější kroky vývoje, vytvořil jsem krátký dotazník. Jeho součástí bylo 5 jednoduchých otázek. Tento dotazník jsem umístil na některé sociální sítě a na většinu fór, které se zabývají aplikacemi na platformě Android. Na anketu odpovědělo 62 lidí. Většinu účastníků tvoří uživatelé středního věku s alespoň základními znalostmi v oboru informačních technologií, což lze vyčíst z obrázku 4.1. Výběr uživatelů se tedy příliš nezmenšil. Ve svém okolí jsem si vybral několik lidí s větším povědomím o informačních technologiích a také několik uživatelů, kteří nejsou z tohoto oboru. Ty jsem zahrnul do vývoje a získal tak cennou zpětnou vazbu. Rozdělení na tyto skupiny je důležité, protože aplikace obsahuje funkce vyžadující nějaké znalosti z informatiky.



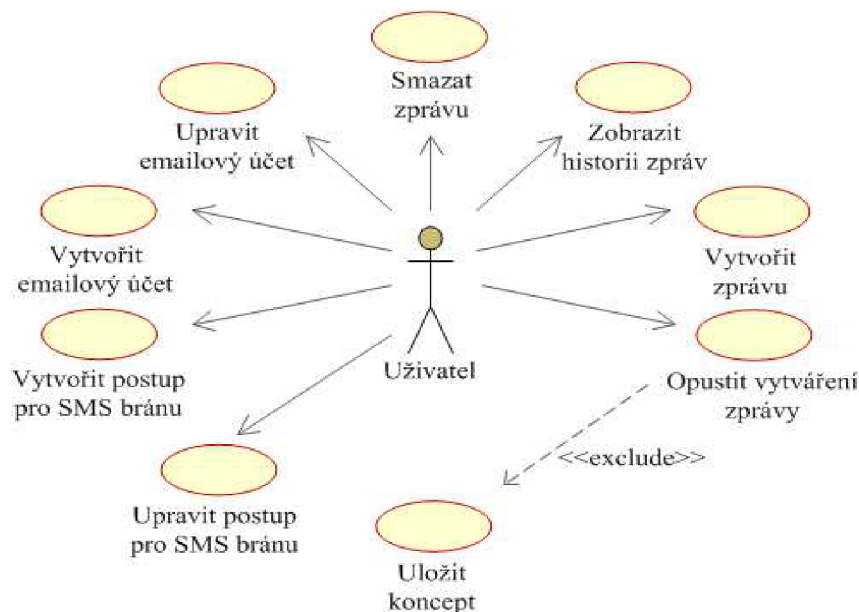
Obrázek 4.1: Jaké jsou Vaše znalosti z oboru informačních technologií?

4.2 Definice funkcí aplikace

První věc, kterou bychom při návrhu měli udělat, je definovat, co všechno bude aplikace umět, co se od ní očekává. Tato informace nám umožní lépe si uvědomit rozsah programu a zaměřit se na jeho nejdůležitější části. Sepsal jsem tedy seznam věcí, které tvoří základ aplikace. Zachycení nejpodstatnějších akcí je v diagramu případu užití na obrázku 4.2. Mezi ně patří především:

- Pohodlné vytváření zpráv pro více lidí.
- Automatické určení internetové brány podle zadaného čísla.
- Přehledné zobrazení historie krátkých textových zpráv a elektronické pošty, jejich správa a řazení.
- Možnost rychlého přeposlání zpráv.
- Ukládání rozepsaných zpráv.
- Potvrzení o odeslání a přijetí zprávy.
- Vytváření a správa účtů elektronické pošty.

- Vytváření a správa informací a postupů pro odesílání zpráv na internetové brány.
- Možnost upravit si aplikaci podle svých preferencí.



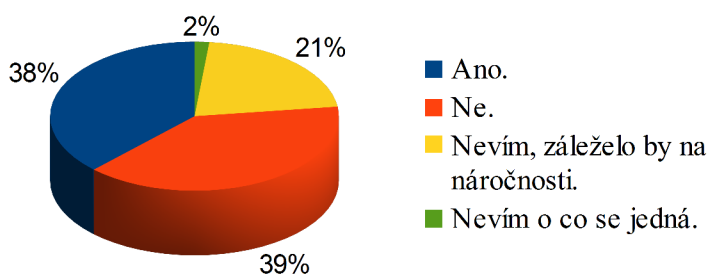
Obrázek 4.2: Diagram případů užití aplikace pro správu textových zpráv.

Upozornění na příchozí zprávy a jejich zobrazení nebylo do programu zahrnuto. Na tomto jsem se dohodl s vedoucím mé bakalářské práce. Aplikace tedy primárně slouží pro odesílání zpráv. Dále vytváření emailových účtů by nemělo uživatele nutit, aby zadával nastavení serveru pošty a další informace spojené s konfigurací účtu. Tvorba účtu je tedy automatizovaná a informace požadované od uživatele zahrnují pouze emailovou adresu a heslo účtu.

Program měl také od začátku umožňovat přidávání postupů pro odesílání krátkých textových zpráv na internetovou bránu. Součástí dotazníku byla otázka týkající se použitelnosti této funkce pro budoucí uživatele, výsledek je na obrázku 4.3. Byl u ní podrobný popis, o co se přesně jedná. Mělo by se tedy počítat s tím, že ne každý je ochotný nebo umí vytvořit skript pro odesílání zpráv přes bránu. S ohledem na to bylo do aplikace zahrnuto několik zabudovaných bran, které dovolují odesílat zprávy přes všechny české operátory.

4.3 Průzkum trhu

Pro platformu Android existuje několik správců krátkých textových zpráv i elektronické pošty. Jako příklad lze zmínit HandCent, Go SMS Pro, Free SMS Sender, Oskárek a MailDroid. Některé z nich jsou propracované a poskytují širokou nabídku funkcí. Část z nich dokonce umožňuje odesílání krátkých textových zpráv přes internetové brány. Také jsem při průzkumu narazil na aplikaci Motorola Universal Inbox, jenž dokonce sjednocuje krátké textové zprávy a elektronickou poštu. Tato aplikace je však nedostupná na mnoha zařízeních. Ovšem nenašel jsem aplikaci, která by poskytovala možnost odesílat SMS zprávy přes internetové brány a navíc dovolovala uživateli přidat si postupy pro odesílání zpráv na brány, které nejsou v aplikaci implementovány. Tedy jejich dodatečné přidání uživatelem. Poslední z uvedených funkcí disponuje program Esemeska, který využívá jazyk JavaScript pro vložení těchto postupů. Tato aplikace je však určena pro stolní počítače a na mobilním zařízení ji nelze využít.



Obrázek 4.3: Vyuzili byste možnost vytvářet skripty pro odesílání na internetové brány?

4.4 Návrh uživatelského rozhraní

Android nabízí ve směru tvorby uživatelského rozhraní velké možnosti a lze vytvořit téměř cokoli. Neklade žádná omezení a nevnucuje jednotný styl vzhledu aplikací. Pouze poskytuje několik rad a doporučení, kterých se můžeme při vývoji držet. Musíme si však uvědomit, že tvorba uživatelského rozhraní na mobilních zařízeních přináší určitá specifika, která je nutné respektovat už ve fázi návrhu. Jedná se především o způsob ovládání a malou obrazovku zařízení. Většina moderních dotykových zařízení má kapacitní dotykový displej, který funguje na principu uzavření elektrického obvodu s objektem, jenž se obrazovky dotýká. Nelze tedy využívat běžná pera pro jeho ovládání, ale pouze vodivé materiály. Nejčastěji se jedná o prst. Tomuto faktu musíme přizpůsobit uživatelské rozhraní aplikace a navrhnout ovládací prvky tak, aby byly snadno dosažitelné. Počet zařízení běžících pod tímto systémem je mnoho a jejich technické vybavení se liší. Musíme tak počítat i s různou velikostí a rozlišením obrazovky. Nemůžeme si tedy dovolit zbytečně plýtvat místem, protože některá zařízení mohou mít malý displej.

Při tomto procesu mi jako vodítko pomohla příručka, která pojednává o vývoji aplikací na platformě Android [8]. Popisuje nezbytné kroky pro definici základních funkcí a vlastností aplikace, začlenění cílové skupiny uživatelů do vývoje, postup pro návrh uživatelského rozhraní a několik tipů pro implementaci.

4.4.1 Tvorba prototypu

Návrh uživatelského rozhraní je důležitý, jelikož nám pomáhá především ve fázi implementace. Máme se tedy čeho držet a můžeme se plně soustředit na programování. V počáteční fázi návrhu jsem vycházel z definovaných funkcí aplikace, které se přirozeně během vývoje ještě měnily, ale základ zůstal stejný. Nakreslil jsem několik skic pouze tužkou. Tady by se žádný pokročilý nástroj ani nehodil, protože se jedná o počáteční návrh. Snažil jsem se vždy vytvořit přibližně deset takových náčrtků a v každém buď původní návrh něčím vylepšit nebo zvolit úplně jiný přístup. Cílem bylo dosáhnout kvalitního výsledku za pomoci většího množství návrhů. Tyto náčrtky jsem pak ukázal několika lidem a spolu jsme vybrali ty nejpodařenější. Přece jen se jednalo o moji první aplikaci pro systém Android a mobilní zařízení vůbec, a tak byl celý cyklus ještě jednou zopakován. Výsledkem byly dva návrhy, se kterými se dále pracovalo. Následovalo převedení skic nakreslených rukou do podoby, která by vypadala lépe. Pro tento účel byl zvolen program Balsamiq, který umožňuje celkem rychlou a pohodlnou tvorbu návrhů.

Dalším krokem bylo z předělaných skic složit funkční prototyp. Jeho výhodou je, že nám dovoluje oživit prezentaci aplikace a předvést její funkce. Je ale nutné, abychom si určili posloupnost jednotlivých akcí, které lze na obrazovce provádět. Prototyp jsem předvedl několika přátelům k posouzení. Umožnilo mi to například zjistit, zda jsou prvky na obrazovkách uspořádány přehledně, jejich význam je pochopitelný a je-li propojení jednotlivých obrazovek vhodné. Byla to také

příležitost pro rekapitulaci možností aplikace, především zda něco důležitého nechybí nebo není přebytečné. Na obrázku 4.4 jsou konečné návrhy hlavních obrazovek uživatelského rozhraní. Šablony, ze kterých se prototyp skládá, jsou většinou určeny pro iPhone. Pro Android jich zatím moc nenajdeme, takže bylo nutné je převzít z jiné platformy.

4.4.2 Popis obrazovek

První obrazovka slouží jako rozcestník. Můžeme se odtud přepnout do historie zpráv, správy emailových účtů a vytvoření nové zprávy. Na druhé šabloně je návrh, jak by mohlo vypadat rozložení prvků pro vytvoření zprávy. Zprávu lze odeslat více lidem, takže bylo nutné navrhnout nějaký prvek pro zobrazení přidaných kontaktů. Ten je řešen jako seznam položek. Všechny akce nad těmito kontakty lze provádět jedním dotekem a jsou tak okamžitě dostupné. Pokud by položky zaplnily celou obrazovku, box pro napsání textu zprávy bude vždy pod nimi a nebude je překrývat. U každého kontaktu lze změnit internetovou bránu, upravit ho a odebrat ze seznamu. Třetí obrazovka obsahuje přehled zpráv, které jsou seskupené podle kontaktu. Návrh pro historii elektronické pošty není mezi šablonami umístěn, protože je v zásadě stejný jako historie SMS zpráv. Liší se pouze tím, že lze přepínat mezi jednotlivými účty a složkami, také zprávy se zde neseskupují podle kontaktu. Další dvě obrazovky slouží pro zobrazení jednotlivých zpráv. Lze nad nimi provádět základní operace, jako mazání, přeposlání a odpovězení. Pokud je zpráva určena pro více lidí, tak jsou vypsána jejich jména a dále je zde šipka pro zobrazení dalších detailů. Poslední šablonu představuje vytvoření nového postupu pro internetovou bránu k odesílání SMS zpráv. Skládá se z prvků pro vložení souborů a zadání doplňujících informací o bráně.

Součástí návrhu je ještě několik obrazovek, které zde nenajdeme. Akce, jež zastupují, nejsou moc frekventované. Nebylo tedy nutné jejich návrh příliš optimalizovat. Jedná se o obrazovky nastavení aplikace a emailového účtu a dále vytváření emailových účtů.

4.5 Návrh tříd aplikace

V systému Android se nabízí strukturovat aplikaci podle návrhového vzoru MVC (Model View Controller). Ten však vede ve výchozím pojetí Androidu ke spojení částí View a Controller do jediné třídy. Výsledkem je třída s velkým množstvím kódu a horší znovupoužitelnost. Pokud chceme dosáhnout lepšího řešení, můžeme využít jiného vzoru nebo lépe navrhnout strukturu MVC. Touto problematikou se zabývá i Framework Android Binding, který nabízí pohodlné využití návrhových vzorů. Využívá konceptu z technologie WPF⁹ a vytváří vazbu mezi prvky uživatelského rozhraní a datovým modelem aplikace. Především se hodí jako podpora pro vzor MVVP (Model View ViewModel). Není to však zadarmo. K programu přidá několik set kilobytů a má také nějaký dopad na efektivitu výsledného programu. Nejvíce mě odradilo, že v době tvorby aplikace neexistovala větší komunita zabývající se touto knihovnou.

9 WPF (Windows Presentation Foundation) - technologie pro práci s uživatelským rozhraním na operačním systému Windows.



Obrázek 4.4: Návrh hlavních obrazovek uživatelského rozhraní aplikace.

Pro tvorbu programu byl nakonec vybrán vzor MVP (Model View Presenter) bez dodatečných knihoven pro jeho podporu. View zde představuje obrazovku (Activity), třídy pro definici ovládacích prvků obrazovky a XML soubor s uživatelským rozhráním. Pro Presenter bude navrženo několik vlastních tříd, které budou obsahovat většinu řídicí logiky. Model je vždy zastoupen třídou, která poskytuje zapouzdření pro přístup k úložišti dat. Vzhledem k tomu, že View není přímo ve spojení s částí Model, jedná se o MVP Passive View. Tento vzor byl zvolen z několika důvodů. Napomáhá při testování aplikace, protože nám dovoluje testovat odděleně jednotlivé její části. Navíc je pro Android přirozenější, než striktní dodržování vzoru MVC. Pomocí něj bychom sice rozdělili aplikaci na nezávislé části, ale ztížili si její implementaci. Úložiště dat

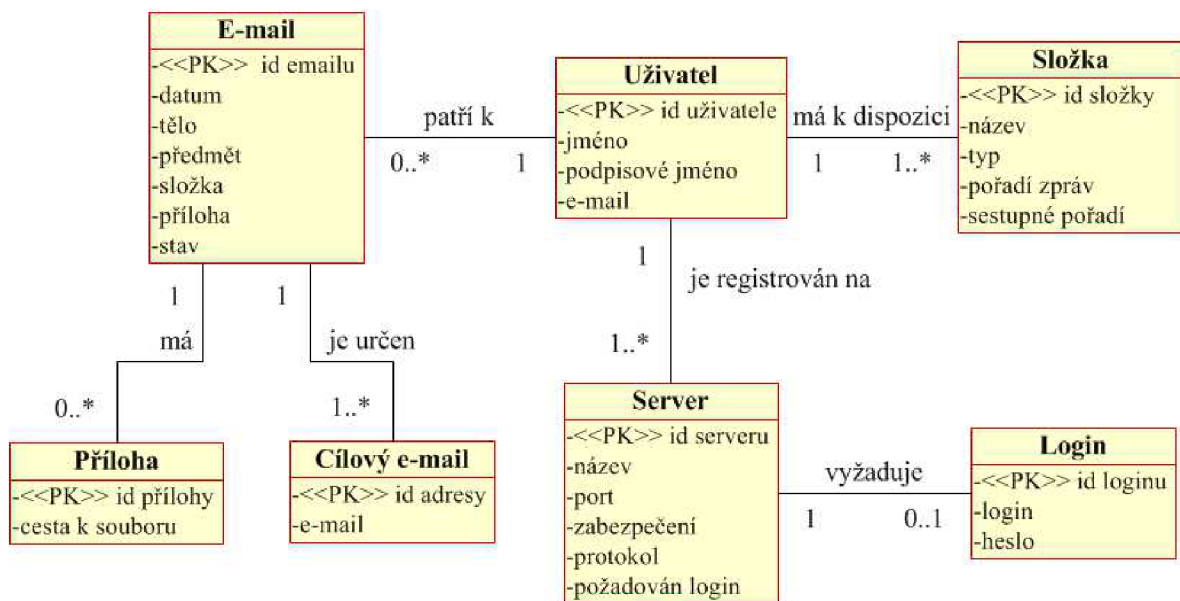
Každá aplikace využívá nějaké místo pro uložení perzistentních dat. Systém Android poskytuje několik typů úložišť, z nichž každé se hodí pro jiný účel. Mezi ty, které jsou vhodné pro vytvářenou aplikaci, patří:

- Velmi důležitým datovým úložištěm, bez něhož se neobejde snad žádná aplikace, jsou preference. Všechny možnosti z nastavení programu se budou ukládat právě sem.

- Máme také k dispozici interní úložiště, kam si aplikace může ukládat soubory pro svoji potřebu a do nějž má přístup pouze ona. Velikost tohoto úložiště je limitována. Vytvářený program sem bude ukládat soubory, které obsahují informace o internetových branách. Toto úložiště bylo vybráno z důvodu omezení přístupu, aby žádný program nemohl měnit obsah těchto souborů. Záleží tak pouze na uživateli, co bude v souborech přítomno. Před vložením souboru musí proběhnout řada kontrol. Například, zda se opravdu jedná o soubor v daném formátu, velikost souboru, která by měla být limitována a další.
- Android poskytuje podporu pro databázi SQLite, která se díky své malé paměťové náročnosti a rychlosti velmi hodí pro mobilní zařízení. Do tohoto úložiště by bylo vhodné ukládat většinu dat spojených s elektronickou poštou a také dodatečné informace o zprávách SMS a branách. Databáze může být i zapouzdřená do rozhraní, tzv. poskytovatele obsahu (Content Provider). Pomocí tohoto poskytovatele mohou všechny aplikace přistupovat k datům v databázi. Do tohoto sdíleného úložiště budou ukládány SMS zprávy.
- A v neposlední řadě je pro naše účely na většině zařízení dostupná paměťová karta, která bude využita pro ukládání příloh elektronické pošty.

4.5.1 Návrh databáze pro uložení elektronické pošty

Program má umožňovat mimo jiné odesílání zpráv elektronické pošty. Proto bylo nutné někam uložit data, která s touto funkcí souvisí. Databáze je pro tento účel vhodná. Dovoluje uložení většího množství dat, která lze rozdělit do tabulek. Navíc lze celý návrh úložiště optimalizovat pro snazší práci s daty a co nejmenší nároky na paměť. Celý datový model je na obrázku 4.5. Jedná se o Entity Relationship diagram, kterým jsou zachycena data „v klidu“. Aplikace sice neslouží pro přijímání zpráv, schéma databáze však bylo navrženo tak, aby případné rozšíření o tuto možnost přineslo co nejméně změn.



Obrázek 4.5: E-R diagram databáze pro uložení dat souvisejících s elektronickou poštou.

4.6 Přidání nových internetových bran

Jak už bylo napsáno, program bude uživateli nabízet možnost přidat podporu pro odesílání zpráv na internetovou bránu. Jako reprezentace postupu pro odeslání zprávy byl zvolen interpretovaný programovací jazyk, konkrétně JavaScript. Obsahem skriptu musí být minimálně jedna funkce, kde

bude zapsána posloupnost příkazů zajišťujících připojení k bráně, odeslání zprávy a podle potřeby další věci. Interpretovaný jazyk byl zvolen z několika důvodů. Především v něm lze vytvořit téměř jakýkoli postup pro odeslání zprávy, což je s ohledem na nestejně rozhraní pro připojení k bránám důležité.

Musíme také mít k dispozici dodatečná data, která jsou s bránou spojená. Každá brána podporuje odesílání zpráv do předem určených zemí a daným operátorům. Dalšími nezbytnými informacemi jsou například maximální délka zprávy, číslo odesílatele, přihlašovací jméno a heslo. Mezinárodní telefonní předvolby, podporované prefixy čísel a délka zprávy budou součástí souboru formátu JSON. Důvodů pro tento způsob uložení je několik. Všechna tato data musíme znát při vytváření zprávy a tak je žádoucí mít je pohromadě. Součástí skriptu nejsou tyto informace zejména kvůli rychlosti načítání ze souboru. Také možnost zadávat je přímo do vstupního pole přes rozhraní aplikace by byla značně nepohodlná, jelikož brána obvykle podporuje desítky prefixů čísel. Formát JSON byl zvolen pro svoji úspornou syntaxi v porovnání například s XML.

4.7 Výběr verze systému

Před tím, než začneme programovat, je nutné určit, pro jaké verze systému bude aplikace dostupná. Pokud bychom toto rozhodnutí nechali až na fázi implementace, mohlo by se stát, že by bylo nutné některé části přepsat. Je to způsobeno tím, že některé verze nejsou plně kompatibilní. Jako minimální podporovaná verze byla zvolena Android Froyo 2.2. Jelikož Android 2.1 a nižší tvoří pouze malou část z celkového zastoupení, bude aplikace dostupná většině uživatelů. Dále můžeme specifikovat verzi, na které jsme primárně program testovali. Pokud by byla aplikace spuštěna na jiné verzi, pak by operační systém poskytl našemu programu prostředí pro zaručení kompatibility. Také je zde ještě jeden atribut, kterým můžeme upřesnit maximální podporovanou verzi, na niž bude náš program fungovat. Všechny tyto atributy jsou umístěny v souboru Android Manifest, který obsahuje základní informace o aplikaci. Tyto modifikátory jsou důležité, pokud chceme naši aplikaci zveřejnit pomocí služby Google Play. Jak už bylo dříve uvedeno, nabízí tato služba aplikace pro platformu Android. Uživatelům se zobrazí v nabídce programů pouze ty, jejichž minimální podporovaná verze systému je rovna nebo menší, než pod kterou pracuje operační systém jejich zařízení.

5 Implementace

Tato kapitola popisuje implementaci některých částí programu. Aplikace je vytvořena pomocí nástroje Android SDK v programovacím jazyce Java. Převážná část uživatelského rozhraní je implementována podle zvyklostí na této platformě, tzn. s využitím jazyka XML.

5.1 Optimalizace aplikace

Protože bude aplikace spuštěna na mobilním zařízení s omezenými zdroji, bylo nutné některé kritické části lépe optimalizovat. Především se jedná o operace, které jsou náročné na čas. V systému Android je zajištěno, že operace probíhající v uživatelském vlákně déle než pět sekund, mohou zapříčinit vynucené zastavení aplikace. Pokud tedy takový případ nastane, je uživateli zobrazena nabídka, zda chce program ukončit. Mezi úkony, které by měly vždy probíhat ve vlákně na pozadí, patří:

- Přístup k databázi, především pak čtení většího množství dat, jejich zápis a úprava.
- Komunikace prostřednictvím sítě.
- Ostatní operace náročné na výpočetní výkon.

Další možností jak snížit nároky aplikace na výkon bylo vytvoření vrstvy mezi databázemi a aplikací. Jsou zde uchovávány data, ke kterým se často přistupuje. Jedná se například o kontakty a ikony náležící ke kontaktům. Při uchovávání obrázků si však musíme dát pozor, abychom nepřesáhli velikost paměti, kterou má aplikace k dispozici. Je tedy dobré omezit velikost takového úložiště. Nabízí se použití referencí na obrázky, které by nebyly pevné. Tedy tzv. `SoftReference` a `WeakReference`. V novějších verzích systému Android je však správce paměti (Garbage collector) více agresivní. Objekty jsou odstraněny z paměti celkem brzy bez ohledu na to, jaký typ slabých referencí používáme. Jejich využití tedy trochu ztrácí význam.

Dále jsem se snažil o co největší integraci se systémem. Jako příklad lze uvést ukládání krátkých textových zpráv do sdílené databáze. Tyto zprávy jsou tedy dostupné i z ostatních programů. Také je zde možnost spustit aplikaci z jiných aplikací, jedná se především o psaní zprávy. Pokud tedy ve správci kontaktů zvolíme, že chceme odeslat někomu krátkou textovou zprávu nebo email, mezi aplikacemi nabídnutými pro provedení akce bude i tato.

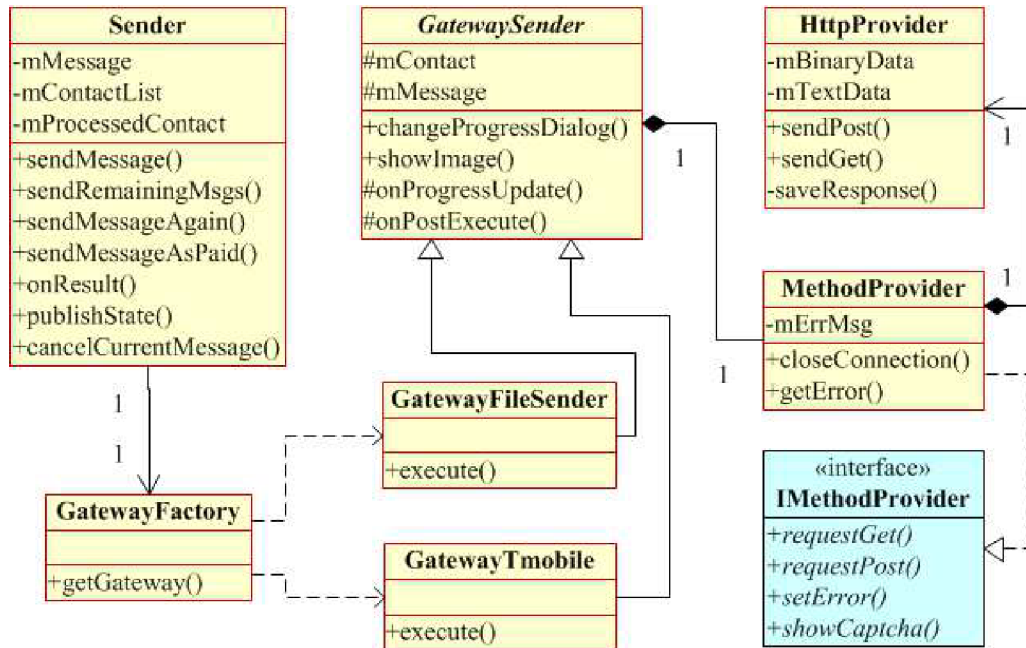
5.2 Odesílání zpráv přes internetové brány

Vzhledem k množství požadavků na odesílání zpráv přes internetové brány, budou třídy této části detailněji rozebrány. Mezi funkce, které je nutné zajistit, patří:

- Snadná rozšiřitelnost o nové brány, které by byly v aplikaci zabudovány.
- Podpora pro zpracování skriptů internetových bran.
- Zobrazení dat uživateli - jedná se především o informace spojené s průběhem odesílání zprávy a také zobrazení kontrolního obrázku.
- Možnost zrušit odesílání zprávy v libovolném okamžiku.

Diagram tříd je na obrázku 5.1. Třída `Sender` obsahuje vše potřebné pro odesílání zprávy více lidem. Uchovává aktuální stav, počet úspěšně odeslaných zpráv a také všechny kontakty, kterým má být zpráva odeslána. Součástí struktury je návrhový vzor továrna (`Factory`), díky němuž můžeme vytvořit potřebnou instanci internetové brány až za běhu programu. Všechny brány mají společného předka `GatewaySender`, který obsahuje zejména metody pro komunikaci s uživatelem. Třídy pro jednotlivé brány obsahují pouze metodu pro připojení, odeslání zprávy a další vyžadované operace.

Takto navržená struktura tříd umožňuje celkem snadno rozšířit program o další vestavěné brány. Další třídou je `MethodProvider`, která zprostředkovává metody pro připojení k internetu. Poskytuje také model pro blokování vlákna do té doby, než uživatel provede nějakou akci. Pokud uživatel odeslání přeruší, v této třídě se požadavek detekuje a operace je zrušena. Tato funkce je implementována pomocí výjimky. Je to jediný způsob, jak přerušit vykonávání skriptu a současně je to i řešení pro vestavěné brány v jazyce Java. Rozhraní `IMethodProvider` je podmnožina metod, které lze při odesílání zpráv využít. Program obsažený ve skriptu může volat metody tohoto rozhraní a ostatní ze třídy `MethodProvider` jsou mu skryty.



Obrázek 5.1: Diagram tříd pro odesílání zpráv přes internetové brány.

5.2.1 Určení internetové brány podle čísla

Aby byla uživateli ulehčena práce při psaní krátké textové zprávy, byla do programu přidána podpora pro automatické určení brány podle zadaného telefonního čísla. Tuto funkci zastává třída `GatewayArbiter`. Vhodná internetová brána pro dané číslo se určí podle následujícího postupu:

- Pokud jsme již na dané číslo nějakou zprávu úspěšně odeslali, pak se zvolí stejná internetová brána jako v minulosti. Tímto se předchází situaci, kdy si někdo změní operátora, ale nechá si své původní číslo. Kdyby tento mechanismus nebyl implementován, pak by byla pokaždé nabízena stejná brána, která zprávu na toto číslo nebude moci odeslat.
- V případě, že zatím na číslo nebyla odeslána žádná zpráva, projdou se postupně všechny internetové brány a ověří se, zda je podporována mezinárodní telefonní předvolba.
- Dále se porovnává prefix čísla s prefixy čísel, na něž brána dovoluje odeslat zprávu. Tento prefix tedy určuje bránu a tedy i operátora.
- V případě nalezení vyhledávání končí. Při neúspěchu je zvolena výchozí brána, kterou nastavil uživatel.

Pořadí výběru bran není náhodné, ale podléhá několika kritériím. Jako první jsou kontrolovány brány zabudované v aplikaci a pak až ty, které byly přidány do aplikace uživatelem. Také jsou upřednostňovány brány, na které uživatel nejčastěji odesílá zprávy. Takto seřazené jsou prezentovány uživateli v nabídce dostupných bran.

5.2.2 Skripty internetových bran

Jak již bylo zmíněno, prostřednictvím souborů lze do aplikace přidat podporu pro odesílání zpráv na nové internetové brány. Jedná se o skript v jazyce JavaScript a soubor ve formátu JSON. Pro zpracování skriptu je použit interpret tohoto jazyka, který poskytuje knihovna Rhino. Pro získání dat ze druhého jmenovaného souboru je použita knihovna Gson. Systém Android sice poskytuje knihovnu pro práci s JSON soubory, ale ta zachovává chybné chování původní knihovny org.json. Ve vyšší verzi systému je tato chyba již opravena. Knihovna Gson je ve verzi s proudovým zpracováním souboru velmi malá a pro aplikaci nepředstavuje žádnou zátěž.

Ve skriptu jsou dostupné dvě globální proměnné, pomocí kterých můžeme komunikovat se zbytkem programu. Jedná se o objekty `MProvider` a `MInfo`. První z nich implementuje rozhraní `IMethodProvider`, které poskytuje metody pro odesílání HTTP požadavků, komunikaci s uživatelem, nastavení chybových a ladících zpráv a další. Druhým z dostupných objektů je `MInfo`, jenž obsahuje data nutná pro odeslání zprávy. Patří sem číslo odesílatele a příjemce, přihlašovací jméno, heslo a text zprávy. Výčet všech metod a atributů těchto objektů můžete nalézt na CD v příloze 1.

5.3 Vytvoření grafického uživatelského rozhraní

Při tvorbě uživatelského rozhraní aplikace byla využita doporučení, která skupina vývojářů systému Android v oblasti designu poskytuje. Jedná se o tzv. grafické návrhové vzory. Jejich úkolem je přinést uživateli co nejvíce pohodlí při používání aplikace a také alespoň trochu sjednotit vzhled a strukturu uživatelského rozhraní na systému Android.

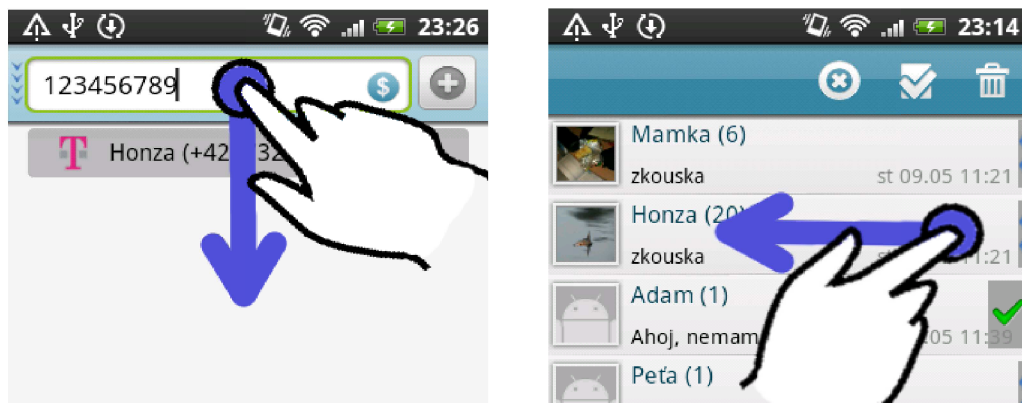
Jedním z těchto vzorů je hlavní nabídka aplikace (Dashboard). Jejím úkolem je poskytnout přehled klíčových funkcí aplikace a také navigaci do těchto částí. Je to tedy první obrazovka, kterou uživatel po otevření programu uvidí. Lze ji vytvořit celkem snadno pouze vhodným uspořádáním prvků na obrazovce. Dále byl využit panel akcí (Action bar), který nabízí nejčastěji využívané možnosti. Ty jsou stále viditelné a okamžitě dostupné. Tento panel je dostupný v systému Android od verze 3.0, pro nižší verze musíme tento prvek vložit dodatečně. V projektu je využita knihovna `Android-actionbar`. Je to velmi malá a jednoduchá knihovna, což jí neubírá na použitelnosti. Naopak lze její funkce snadno přizpůsobit a optimalizovat pro náš program. Posledním ze vzorů, které byly při tvorbě využity, je upravené kontextové menu (Quick actions). Může jím nahradit standardní kontextovou nabídku a také poskytnout přehledné menu pro další akce. Opět však musíme tuto funkci obstarat sami. Knihovna `NewQuickAction3D` nám nabízí vše podstatné a vzhled nabídky lze do značné míry upravit. Velikost knihovny je malá, jelikož zahrnuje pouze pár nepříliš objemných tříd.

5.3.1 Ovládání pohybem prstu

Ovládání aplikace pomocí gest, tedy táhnutím prstu, může přispět ke zlepšení uživatelské přívětivosti. Pro zařízení s malým displejem představuje tato funkce další výhodu, protože může ušetřit místo na obrazovce. Použití této možnosti vyžaduje, aby funkce, kterou zastupuje, byla z hlediska použití intuitivní. Na obrázku 5.2 je ukázka použití gest v aplikaci. Gesta byla využita zejména z důvodu větší přehlednosti a omezení počtu ovládacích prvků na obrazovce. Tato možnost byla přidána až ve fázi implementace, v návrhu tyto prvky tedy nenajdeme. Na některých místech jsou zobrazovány bubliny s nápovědou, aby uživatel tento způsob ovládání nepřehlédl. Zobrazení nápověd se dá v nastavení vypnout, takže zbytečně neruší. Také jsou zde grafické prvky, jejichž účelem je napovědět, co je možné udělat a jak.

Pro detekci tahu prstem je použit `GestureDetector`, který nám předá informace o provedeném gestu prostřednictvím třídy `GestureListener`. Můžeme specifikovat typ gest, na

kteřá chceme reagovat. Patřĩ sem například tažení, poklepání a podržení prstem. Pro gesto tažení lze určit požadovaný směr, maximální a minimální rychlost a délku pohybu.



Obrázek 5.2: Příklad gest, která je možné použít v aplikaci.

6 Testování aplikace

Tato kapitola popisuje především způsob, jakým bylo provedeno testování aplikace na uživatelích. Probíhalo to tak, že uživatelé měli vypracovat několik úkolů. Po provedení každého úkolu následovaly otázky, které měly testovanou část programu zhodnotit. Otestování, zda aplikace funguje správně, bylo provedeno také. Zejména bylo nutné vyzkoušet program na větším počtu rozdílných zařízení a verzí systému Android. Tyto testy byly částečně provedeny při testování aplikace uživateli, jelikož vypracování úkolů probíhalo na jejich zařízení. Testy z tohoto hlediska jsou nezbytné především z důvodu omezené kompatibility jednotlivých typů zařízení i verzí systému. Aplikace také byla testována na systémech Android od verze 3.0 a výše, ale pouze na emulátoru.

Aby bylo dosaženo co nejhodnotnější odezvy od uživatelů, byli vybíráni lidé z různých věkových kategorií a profesí. Bylo také důležité mít několik uživatelů, kteří jsou z oboru informačních technologií a mohli by zhodnotit i pokročilé funkce aplikace. Podmínkou však bylo, že musí vlastnit telefon se systémem Android nebo alespoň mít s ním nějaké zkušenosti.

6.1 Úkoly k vypracování

Testování probíhalo osobně, pokud to bylo možné. Pro vypracování úkolů, u kterých jsem nebyl přítomen, byl využit elektronický dotazník. Uživateli byl na začátku vysvětlen účel aplikace. Pak se přistoupilo k samotnému testování. Vždy byl nejdříve předložen jeden úkol, který měl uživatel provést. Pak následovalo několik otázek, jejichž odpovědi byly většinou formou výběru jedné odpovědi ze skupiny možných. Při vyplňování odpovědí byly uživatelům kladeny doplňující otázky, například: „Jak by jsi chtěl, aby toto bylo udělané, co by ti při tomto úkolu usnadnilo práci“. V dalších odstavcích je uvedeno znění úkolů a také jejich význam.

6.1.1 Vytvoření zprávy

Tento test má za cíl vytvořit novou krátkou textovou zprávu třem lidem a projít možnosti pro manipulaci s kontakty. Uživatel si zde vyzkouší vytváření zpráv a také může porovnat provedení této funkce s jinými aplikacemi, které sám používá. Postup pro jeho splnění je následující:

- Přepnout se do obrazovky pro psaní SMS zpráv.
- Přidat kontakty, kterým bude odeslána zpráva. První kontakt bude vybrán pomocí tlačítka vpravo pro přidávání kontaktů.
- Druhé číslo je nutné napsat ručně, nemělo by být uloženo v telefonu (lze zadat například 000000000). Dále se pokusit přidat tento kontakt do seznamu kontaktů, kterým bude odeslána zpráva.
- Poslední číslo vybrat tak, že do boxu bude napsáno jméno kontaktu a následně vybráno z nabídky.
- Zkusit změnit internetovou bránu, přes kterou se bude zpráva odesílat.
- Odstranit číslo, které bylo zadáno jako druhé (obsahuje samé nuly).
- Nakonec napsat libovolný text a zprávu odeslat.

Uživateli byly po provedení všech bodů úkolu předloženy tyto otázky:

- Vyhovoval Vám způsob, jakým se kontakty přidávají do seznamu kontaktů, kterým bude odeslána zpráva?
- Bylo pro Vás pochopitelné, že kontakty lze přidávat do seznamu přidávaných kontaktů pohybem prstu od boxu pro psaní zprávy směrem dolů?

- Přejde Vám způsob zobrazení přidaných kontaktů optimální nebo byste uvítali formu, která by nezabírala tolik místa na obrazovce?

6.1.2 Zobrazení historie zpráv

Cílem úkolu je projít si historii zpráv a vyzkoušet si některé možnosti pro práci s nimi. Tento test má především zjistit, zda je zobrazení jednotlivých zpráv přehledné a podání možností pro jejich správu pochopitelné. Úkol je následující:

- Přepnout se do výpisu krátkých textových zpráv.
- Zobrazit zprávu, která byla v předchozím úkolu odeslána.
- Zkusit smazat tuto zprávu.

Po tomto jednoduchém úkolu následovalo několik otázek:

- Bylo pro Vás zobrazení zpráv přehledné?
- Na pravé straně položky zprávy je šedý proužek, který slouží pro výběr zpráv. Jednotlivé zprávy lze vybírat přetažením prstu zprava doleva. Byla tato možnost manipulace se zprávami jasná na první pohled?

6.1.3 Manipulace se zprávami

Zde si uživatel vyzkouší výběr a manipulaci se zprávami pomocí gest. Lze tak zjistit, jaký přínos má použití gest pro hromadné vybírání zpráv, které lze potom smazat nebo přesunout. Tato možnost je postavena proti klasickým zaškrtačacím polím a kontextové nabídce. Postup pro vypracování testu je tento:

- Přepnout se do výpisu krátkých textových zpráv.
- Vybrat několik zpráv přetažením prstu zprava doleva.
- Projít si možnosti, které jsou nad vybranými zprávami k dispozici.
- Zkusit si smazat některou ze zpráv.

Dále byly uživateli předloženy tyto otázky:

- Myslíte si, že je tento způsob manipulace se zprávami výhodný? Využívali byste takovýto postup?
- Vyhovovala Vám manipulace se zprávami pomocí přetahování prstem, nebo byste raději měli na řádku stále viditelné zaškrtačací pole?

6.1.4 Dodatečné otázky

Na závěr bylo přichystáno pár otázek týkajících se celé aplikace. Především se jednalo o její zhodnocení z hlediska provedení a nabízených funkcí. Odpovědi na tyto otázky byly písemné. Zde je jejich seznam:

- Je nějaká funkce, kterou používáte ve stávajícím správci zpráv a ocenili byste, kdyby byla i tady?
- Co Vám v aplikaci chybělo nebo naopak bylo na obtíž?
- Přejde Vám sloučení emailů a krátkých textových zpráv do jediné aplikace jako dobrá funkce?

6.2 Výsledky testování

Odpovědi na otázky z dotazníků, které měly formu výběru z několika možností, byly sečteny a zaneseny do grafů. Ty můžete nalézt v příloze 2. Textové odpovědi byly ukládány a dále

zpracovávány ručně. Nasbírané výsledky posloužili jako dobrý zdroj pro opravu a také jako návrhy pro vylepšení aplikace. Většina uživatelů byla s drobnými výhradami s navrženým prostředím spokojena. Někdo si po procesu testování aplikaci v telefonu ponechal a nadále ji používá.

K celkovému grafickému zpracování aplikace neměl nikdo z dotázaných výhrady. Sloučení krátkých textových zpráv a emailů se setkalo s celkem kladným ohlasem. Někomu přišlo dobré, že je zde jednotné rozhraní pro vytváření zpráv a také že se nemusí přepínat do jiné aplikace. Asi polovině z dotázaných však trochu vadilo, že aplikace neumožňuje přijímání zpráv a ani jejich zobrazení. Tato poznámka byla přijata jako námět pro rozšíření aplikace. Následuje rozebrání několika poznatků, které testy přinesly.

6.2.1 Test pro vytvoření nové zprávy

Většině uživatelů způsob práce s kontakty vyhovoval. Použití gest pro tento účel jim přišlo jako dobrý nápad a používali by to. Asi čtvrtině uživatelů připadala nápověda pro přidávání kontaktů pomocí gest málo výrazná. Všichni se však shodli na tom, že zobrazení kontaktů je vhodné a neměnili by jej za úspornější formu.

6.2.2 Test pro zobrazení historie zpráv

Téměř všichni uživatelé se ve výpisu zpráv orientovali a byl pro ně přehledný. Jen k jedné věci mělo hodně lidí výhrady, a to k zobrazení krátkých textových zpráv, které patří ke konverzaci. Přišlo jim, že jednotlivé zprávy splývají a špatně se mezi nimi orientuje. Proto jsem tuto část ještě upravil a pozadí zpráv barevně odlišil. Pár testujícím osobám nebylo vlastní, že výpis krátkých textových zpráv začíná shora a nikoli odspodu, jak je to ve výchozím správci. Dále asi polovina dotázaných si možnosti výběru zpráv pomocí gest nevyšimla. Raději by zvolili výraznější formu, která by klidně mohla zabírat více místa na obrazovce.

6.2.3 Testování manipulace se zprávami

Způsob manipulace se zprávami pomocí gest se všem uživatelům líbil a používali by ho. Jen by asi polovina volila výraznější podobu nápovědy. Nikdo z dotázaných by jej neměnil za přímo viditelné zaškrtačkové pole. Všem také přišlo výhodnější, že nejsou nuceni pracovat s každou zprávou zvlášť, jako je tomu při používání kontextové nabídky.

7 Závěr

Cílem této práce bylo vytvoření aplikace pro správu textových zpráv na platformě Android. Tedy aplikace pro odesílání a práci s krátkými textovými zprávami a elektronickou poštou.

Byl vyvinut program, který umožňuje odesílat zprávy různou formou. Jednak jsou to krátké textové zprávy, které je možné odeslat standardní cestou přímo přes mobilního operátora. Také můžeme využít služeb zprostředkovatele, v tomto případě se jedná o internetovou bránu. Pro vypracování aplikace bylo nutné být obeznámen se strukturou mobilní sítě a především se spoluprací této sítě s internetovými branami. Program dále nabízí možnost vložení postupu pro odeslání zprávy internetové bráně, který v aplikaci není obsažen. Bylo nutné zvolit vhodnou formu pro tento postup vzhledem k faktu, že mobilní zařízení má omezené zdroje. S touto funkcí je spojené i zajištění bezpečnosti uživatele, že postup nebude obsahovat kód, který by například odesílal citlivé informace někomu jinému. Uživatel tak má možnost přidat si podporu pro odesílání zpráv přes oblíbenou bránu, u které může mít vytvořený účet. Dále lze odesílat elektronickou poštu. U obou forem jsou podporovány například funkce pro přeposlání, potvrzení o přijetí, potvrzení o doručení a také odesílání zprávy více lidem. Uživatelské testování ukázalo, že aplikace se dá ovládat velmi snadno a nabízí několik vylepšení, které její používání zpříjemňují. Cíl této bakalářské práce byl tedy splněn. Program je dostupný veřejnosti prostřednictvím služby Google Play.

Vypracování této práce pro mě mělo velký přínos. Získal jsem přehled v oblasti moderních mobilních zařízení a také zjistil, jaká specifika přináší vývoj aplikací pro tato zařízení. Především jsem se však naučil vytvářet programy pro mobilní platformu Android, která je v dnešní době velmi rozšířená. Také jsem nabyt zkušenosti s jazykem Java, se kterým jsem se podrobněji seznámil až při této práci. Při psaní dokumentace jsem se snažil, aby poskytla čtenáři mimo jiné návod pro tvorbu aplikací pro systém Android. Jsou zde rozebrány fáze analýzy, návrhu a implementace uživatelského rozhraní a také ostatních částí aplikace.

Jako rozšíření programu by bylo možné přidat funkci pro přijímání zpráv. A to jak krátkých textových, tak i elektronické pošty. Dalším možným rozšířením je dovolit uživateli ukládat postupy, které slouží pro odesílání zpráv na internetové brány, na server. Tam by byly veřejně dostupné a každý uživatel by si tak mohl najít a přidat do programu podporu pro novou bránu. Toto rozšíření by nabylo většího významu, kdyby aplikace byla rozšířena ve více zemích. Program by pak obsahoval hlavně podporu pro ty brány, které jsou pro uživatele dosažitelné.

Literatura

- [1] RICHTR, Tomáš. Základní struktura sítě GSM. *Technologie pro mobilní komunikaci* [online]. 2002 [cit. 2012-04-15]. Dostupné z: <<http://tomas.richtr.cz/mobil/gsm-strukt.htm>>
- [2] BARR, Dale, FONASH, Peter. SMS over SS7 [online]. 12/2003 [cit. 2012_04_20]. Dostupné z: <http://www.ncs.gov/library/tech_bulletins/2003/tib_03-2.pdf>
- [3] ETSI. ETSI TS 123 038. *Digital cellular telecommunications system (Phase 2+) Universal Mobile Telecommunications System (UMTS); LTE; Alphabets and language-specific information*. Version 10.0.0 Release 10. 2011-03. Dostupné z: <http://www.etsi.org/deliver/etsi_ts/123000_123099/123038/10.00.00_60/ts_123038v100000p.pdf>
- [4] HILLEBRAND, Friedhelm. *Short Message Service (SMS): The Creation of Personal Global Text Messaging*. 2. vyd. 2010. ISBN 978-0470688656.
- [5] KLENSIN, J. AT&T LABORATORIES. RFC 2821. *Simple Mail Transfer Protocol*. April 2001.
- [6] JVM Scripting Languages. MURPHY, Mark. *The Busy Coder's Guide to Advanced Android Development*. United States of America: CommonsWare, 2009, s. 451. ISBN 978-0-9816780-1-6.
- [7] Rhino History. *Mozilla* [online]. MOZILLA, ©1998–2012 [cit. 2012-05-01]. Dostupné z: <<http://www.mozilla.org/rhino/history.html>>
- [8] STEWART, Kyle. *Android Human Interface Guidelines* [online]. 19.08.2009, 35 s. Dostupné z: <<https://island.byu.edu/files/AndroidHIG.pdf>>
- [9] Activity. *Android developers* [online]. Google, 01 May 2012 [cit. 2012-04-20]. Dostupné z: <<http://developer.android.com/reference/android/app/Activity.html>>
- [10] User Interface Definition. The Linux Information Project [online]. December 2, 2005 [cit. 2012-05-01]. Dostupné z: <http://www.linfo.org/user_interface.html>
- [11] ELGIN, Ben. Google Buys Android for Its Mobile Arsenal. Bloomberg Businessweek [online]. AUGUST 17, 2005 [cit. 2012-05-07]. Dostupné z: <http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm>
- [12] Android: nová platforma pro mobilní zařízení. *T-Mobile: Tiskové centrum* [online]. T-MOBILE, 07.11.2007 [cit. 2012-05-15]. Dostupné z: <http://t-press.cz/tiskove_zpravy/2007/770/>
- [13] Signing Your Applications. *Android developers* [online]. Google, 01 May 2012 [cit. 2012-05-02]. Dostupné z: <<http://developer.android.com/guide/publishing/app-signing.html#secure-key>>
- [14] What is Android?. *Android developers* [online]. Google, 01 May 2012 [cit. 2012-05-02]. Dostupné z: <<http://developer.android.com/guide/basics/what-is-android.html>>
- [15] Tools. *Android developers* [online]. Google, 01 May 2012 [cit. 2012-04-30]. Dostupné z: <<http://developer.android.com/guide/developing/tools/index.html>>
- [16] What is the NDK?. *Android developers* [online]. Google, [01 May 2012] [cit. 2012-05-02]. Dostupné z: <<http://developer.android.com/sdk/ndk/overview.html>>
- [17] Project Home. *Android-scripting: Scripting Layer for Android brings scripting languages to Android*. [online]. [01 May 2012] [cit. 2012-05-02]. Dostupné z: <<http://code.google.com/p/android-scripting/>>
- [18] Android API Levels. *Android developers* [online]. Google, [2012], 01 May 2012 [cit. 2012-05-02]. Dostupné z: <<http://developer.android.com/guide/appendix/api-levels.html>>

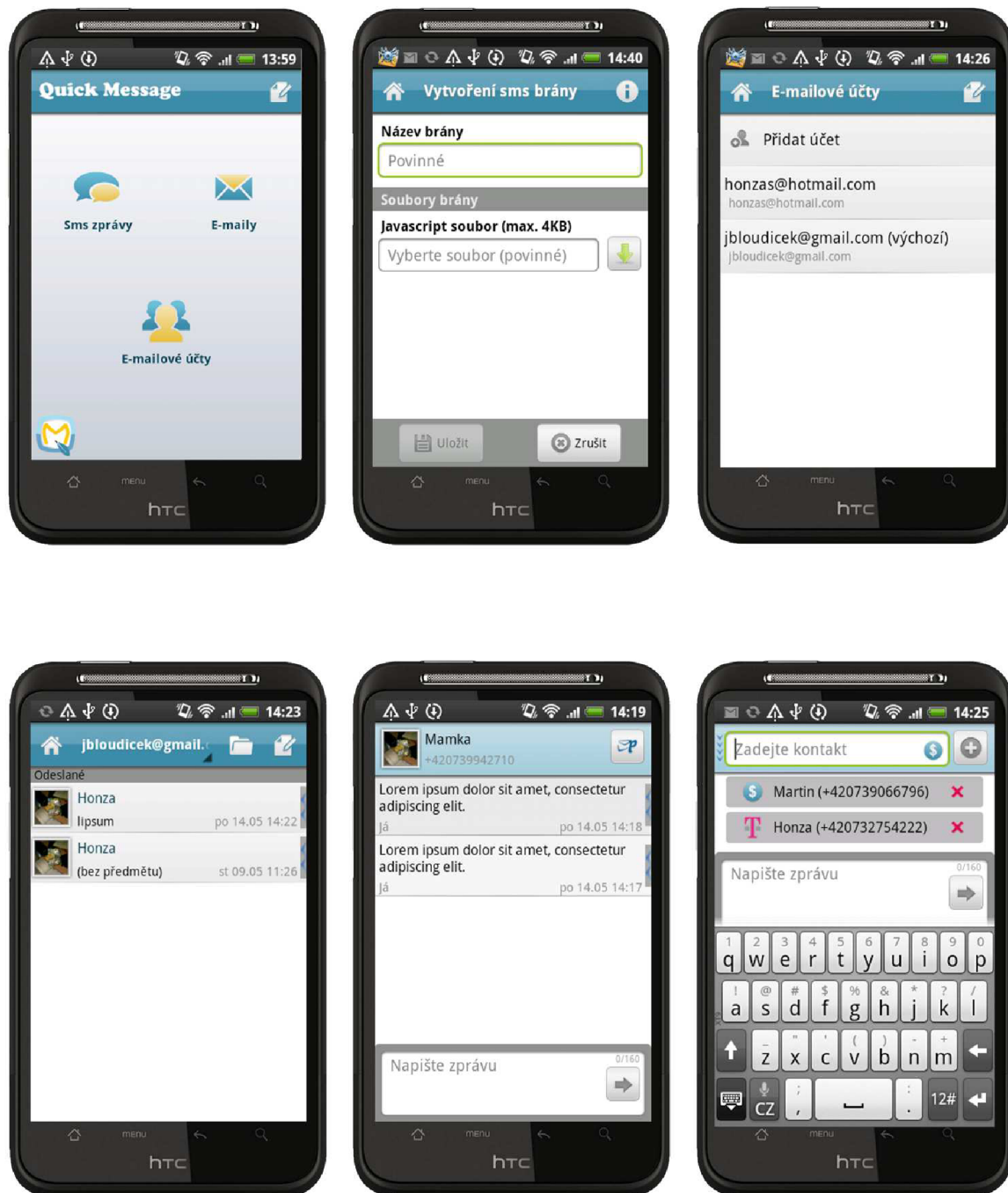
Seznam příloh

Příloha 1. Obrazovky aplikace

Příloha 2. CD

- Zdrojové kódy aplikace: /src/
- Manuál k aplikaci: /manual/manual.pdf
- Grafy z uživatelského testování aplikace: /tests/graphs.pdf
- Text bakalářské práce ve zdrojovém tvaru: /thesis/openoffice/bp.odt
- Text bakalářské práce ve formátu PDF: /thesis/pdf/bp.pdf

Příloha 1: Obrazovky aplikace



Obrázek přílohy 1.1 : Náhledy hlavních obrazovek aplikace.