

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

POKROČILÉ POZNÁMKY PRO ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL MÁCA

BRNO 2013



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

POKROČILÉ POZNÁMKY PRO ANDROID

ADVANCED NOTES FOR ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL MÁČA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RUDOLF KAJAN

BRNO 2013

Abstrakt

Tato práce se zabývá návrhem a implementací aplikace pro práci s poznámkami na operačním systému Android. Aplikace využívá senzory mobilní platformy pro pokročilejší funkce. Úvodní část práce se zabývá teoretickými informacemi o operačním systému Android a informacemi o tvorbě aplikací pro tento operační systém. Praktická část obsahuje návrh aplikace, její implementaci a testování.

Abstract

This thesis deals with a proposal and an implementation of an application for work with notes on the Android operating system. This application uses the sensors of a mobile platform for advanced functions of the application. The first part of thesis deals with the theoretical information about the Android operating system and information about application development for this operating system. The theoretical part contains proposal of the application, her implementation and the testing.

Klíčová slova

Android OS, poznámky, GPS, WIFI, GSM, sémanticky důležitá místa, Google maps

Keywords

Android OS, notes, GSP, WIFI, GSM, semantically meaningful places, Google maps

Citace

Michal Máca: Pokročilé poznámky pro Android, bakalářská práce, Brno, FIT VUT v Brně, 2013

Pokročilé poznámky pro Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Rudolfa Kajana.

.....

Michal Máca
3. května 2013

Poděkování

Zde bych rád poděkoval mému vedoucímu bakalářské práce panu Ing. Rudolfu Kajanovi za konzultace a odbornou pomoc. Dále bych rád poděkoval svojí rodině za podporu při studiu a všem, kteří se podíleli na testování.

© Michal Máca, 2013.

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

Obsah

1	Úvod	2
2	Existující aplikace	3
2.1	<i>Catch Notes</i>	3
2.2	<i>ColorNote</i>	3
2.3	<i>Evernote</i>	4
2.4	<i>Note Everything</i>	4
3	Systém Android	6
3.1	Historie	6
3.2	Architektura systému Android	6
3.3	Verze systému	8
3.4	Vývoj aplikací	10
3.5	Tvorba uživatelského rozhraní	13
3.6	Lokace v systému Android	14
3.7	Google Maps API	16
4	Návrh aplikace	17
4.1	Uživatelské požadavky	17
5	Implementace aplikace	19
5.1	Části aplikace	19
5.2	Získávání uživatelsky důležitých míst	23
5.3	Automatické otevření poznámky	24
5.4	Ukládání do databáze	24
6	Testování	27
6.1	Testovaná zařízení	27
6.2	Testování uživateli	28
7	Závěr	30
A	Obsah CD	32
B	Testovací úkoly	33

Kapitola 1

Úvod

V dnešní době, kdy jsou na člověka kladeny čím dál větší nároky, je třeba mít možnost zaznamenávat si co nejrychleji a nejpohodlněji jakékoli zápisky či poznámky. Nosit u sebe neustále poznámkový blok a pero již vyšlo z módy a je tedy třeba hledat jiné alternativní řešení. Jako jedno z nejrozumnějších řešení se jeví možnost využití chytrých telefonů pro zaznamenávání poznámek.

Téměř každý sedmý člověk na planetě dnes používá telefon s operačním systémem. K datu 15.12.2012 bylo zaznamenáno již 1038 miliard uživatelů. Nejpoužívanějším operačním systémem v mobilních zařízeních je *Android*, který používá 53 % ze všech uživatelů chytrých telefonů a každým dnem se tento počet zvětšuje o 1,3 milióny. Nejvhodnějším operačním systémem se z hlediska použitelnosti jeví právě operační systém Android, jelikož jeho zastoupení mezi uživateli je největší a jedná se o nejrychleji se rozšiřující platformu.

Vzhledem k současnému postupu ve vývoji technologií na určování polohy by mohlo být zajímavé využít tyto techniky k obohacení vyvíjené aplikace. Uživatel tak bude mít možnost zobrazit pozici kde poznámku vytvořil, případně upravoval a detekovat pro něj sémanticky důležitá místa.

Cílem této práce je nastudovat vývoj aplikací pro operační systém Android a navrhnout aplikaci, která umožní vytvářet poznámky s využitím senzorů mobilní platformy.

První kapitola popisuje již existující aplikace pro tvorbu poznámek, jejich výhody a nevýhody.

Cílem druhé kapitoly je seznámit se s operačním systémem Android, jeho historií, různými verzemi a tvorbou aplikací. Dále je zde popsána možnost získávání dat o poloze a způsob využití *Google Maps API* pro zobrazení map v aplikacích.

Ve třetí kapitole je popsán návrh, implementace a testování vyvíjené aplikace.

Kapitola 2

Existující aplikace

V současné době již existuje celá řada aplikací, které poskytují možnosti pro tvorbu a následnou úpravu poznámek na zařízeních s operačním systémem Android.

V následujících podkapitolách jsou popsány aplikace, jejich výhody a nevýhody, které jsou zdarma ke stažení z *Google Play*¹ a jsou hodnoceny jako jedny z nejlepších pro operační systém Android. S těmito aplikacemi mám také vlastní zkušenosti a své postřehy k nim přikládám níže.

2.1 *Catch Notes*

Tato aplikace mě velmi zaujala hned při prvním spuštění. Má velmi elegantní design, podobný Androidu verze 4.x a velmi intuitivní ovládání. *Catch* umožňuje přidat několik typů poznámek, konkrétně klasické textové, zvukové, *todo* seznam, fotografii jako poznámku nebo časovanou poznámku.

Poznámky lze jednoduše, pomocí barevného označení, přiřadit do různých skupin, které lze různě spravovat. Další velmi šikovnou věcí je možnost sdílet poznámky s přáteli nebo si je sesynchronizovat s Google kalendářem a pracovat s nimi *offline*.

Této aplikaci bych vytkl pouze dvě věci – spuštění trvá na můj vkus příliš dlouho (průměrně tři až čtyři sekundy), jako zásadní nedostatek bych však viděl nemožnost přidat si na plochu jako *widget* libovolnou poznámku a mít ji neustále na očích, popřípadě ji rychle upravovat.

Jak aplikace vypadá je znázorněno na obrázku 2.1.

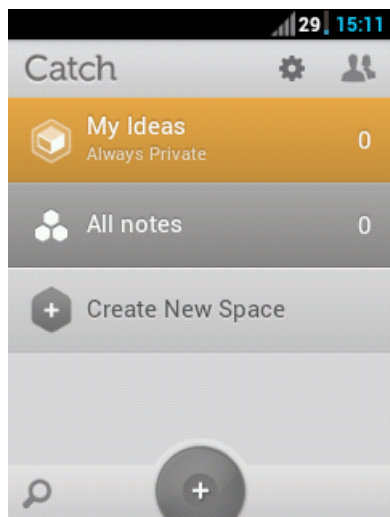
2.2 *ColorNote*

Aplikace *ColorNote* poskytuje oproti *Catch* již možnost vkládat na plochu jednotlivé vytvořené poznámky jako *widgety*. Ovšem následná jejich úprava je poměrně zdlouhavá. Celé aplikaci by se dalo rovněž vytknout poměrně nepříliš intuitivní ovládání. Další nevýhodou oproti *Catch* je nemožnost přidat k poznámce pozici, kde byla vytvořena.

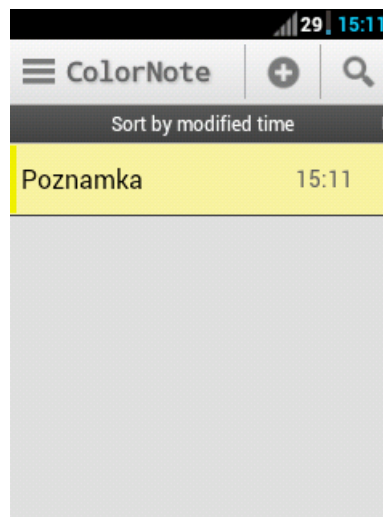
Největší zápor uživatel zjistí v případě, že si na plochu uloží nějaké poznámky, restartuje nebo vypne a následně zapne telefon a místo poznámek se na ploše objeví jen prázdné štítky, které je nutné ručně smazat a následně požadované poznámky znovu na plochu přidat.

Úvodní stránka aplikace s již vytvořenou poznámkou je pro ilustraci na obrázku 2.2.

¹<https://play.google.com/store>



Obrázek 2.1: Catch Notes, verze 5.1.1



Obrázek 2.2: ColorNote, verze 3.9.11

2.3 *Evernote*

Evernote je jeden z nejznámějších a nejrozšířenějších programů pro tvorbu a správu poznámek, a to nejen na mobilních telefonech. Tato aplikace poskytuje možnost vytvářet několik typů poznámek (textové, zvukové, fotografie) a sdílet je nejen mezi mobilními telefony, ale také mezi telefonem a počítačem. Na počítači je klient podporován na *Windows*, *MacOS* a neoficiálně také na *Linux*. K vytvořeným poznámkám je možné navíc přistupovat pomocí webového rozhraní.

V tomto programu je možné třídit poznámky do jednotlivých složek a jednoduše v nich vyhledávat. K aplikaci je možné stáhnout rozšíření (opět zdarma), které poskytuje možnost vkládat na plochu widgety, které umožňují rychlé vytvoření všech typů poznámek a rychlý přístup do aplikace. Opět zde ovšem chybí widget, který by na ploše zobrazoval detail zvolené poznámky. Aplikace také poskytuje možnost přidat k poznámce lokaci, což musí ovšem uživatel provést manuálně (kliknutím na mapu). Po instalaci je nutné se zaregistrovat, k čemuž je vyžadováno internetové připojení, aby mohl uživatel vytvářet poznámky.

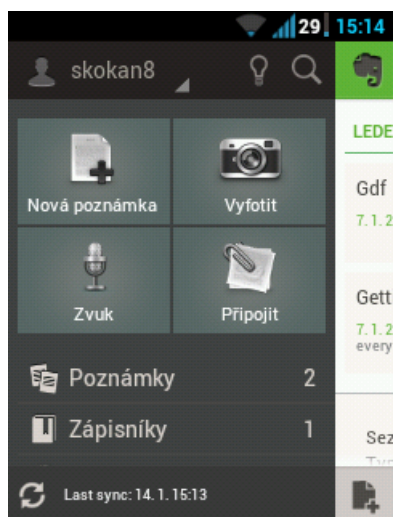
Obrazovka po vytvoření účtu je ilustrována na obrázku 2.3.

2.4 *Note Everything*

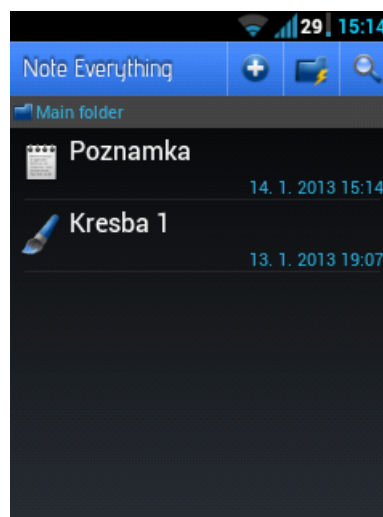
Note Everything poskytuje možnost přidávání textových, zvukových, kreslených poznámek a poznámky z *QR kódů*. Umožňuje třídění jednotlivých poznámek do složek, export poznámek na paměťovou kartu a vyhledávání poznámek. Rovněž poskytuje možnost přidat na plochu několik typů widgetů. Widgety mohou zobrazovat konkrétní poznámky, další variantou je widget pro rychlé prohledávání dříve vytvořených poznámek. Tato aplikace je dostupná ve dvou verzích. Verze zdarma a verze *pro*, která je placená a poskytuje další rozšířené možnosti.

Hlavním záporem téhle aplikace je, že v bezplatné verzi neposkytuje možnost vytvořit poznámku na určité datum a čas a její připomenutí.

Vzhled aplikace je ukázán na obrázku 2.4.



Obrázek 2.3: Evernote, verze 4.4.2



Obrázek 2.4: Note Everything, verze 4.2.8-79

Kapitola 3

System Android

Android není jen operační systém pro mobilní telefony, ale jedná se o *open source* platformu, zejména pro mobilní zařízení (chytré telefony, PDA, ...). Tato platforma zahrnuje operační systém, aplikace, uživatelské rozhraní a *Middleware* (softwarová vrstva ležící mezi operačním systémem a aplikacemi, sloužící k odstínění aplikačního vybavení od základních služeb operačního systému).

3.1 Historie

Obsah této kapitoly je čerpán především z [11] a [3].

Společnost Android Inc. byla založena v roce 2003 čtyřmi lidmi, a to Andy Rubinem, Nickem Searsem, Chrisem Whitem a Richem Minerem. Android Inc. byla poté v roce 2005 převzata společností Google Inc. jako jeho dceřiná společnost.

Google Android pod vedením Andyho Rubina vyvinul platformu založenou na linuxovém jádře. Jedná se o *monolitické jádro*, jehož veškerý kód běží ve stejném adresovém prostoru. O samotný vývoj se ale zasloužilo především uskupení výrobců mobilních telefonů, telekomunikačních operátorů a technologických firem, které jsou pod záštitou Google a nazývají se konsorcium *Open Handset Alliance* (OHA). Toto konsorcium bylo založeno 6. listopadu 2007 a sdružovalo 34 výrobců. Dnes OHA zahrnuje více jak 80 firem a jejich počet se nadále zvětšuje.

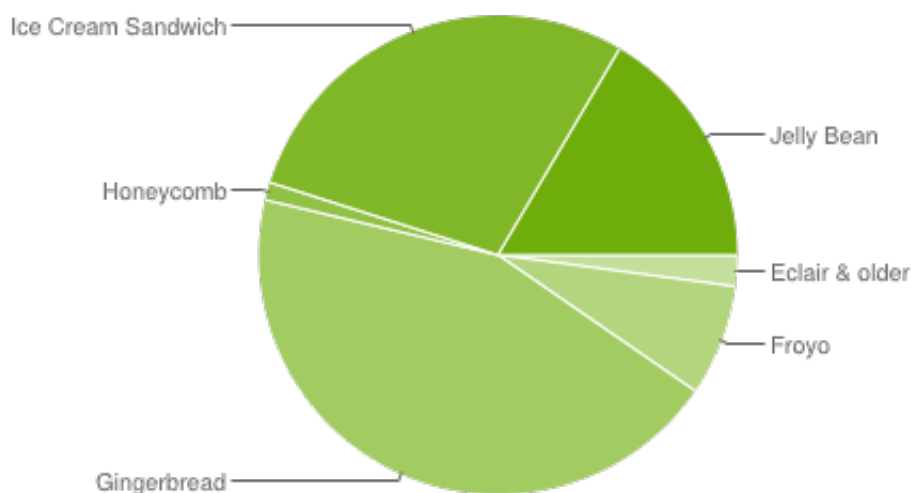
V roce 2008 Google Android představil první oficiální verzi svého systému: Android 1.0 na zařízení *HTC G1 'Dream'*.

Dnes, jak již bylo zmíněno v úvodu, Android využívá více než 50 % uživatelů ze všech, kteří vlastní chytré telefony. Nejrozšířenější verzí je v současné době Android 2.3.x. Na obrázku 3.1 je zobrazena rozšířenost jednotlivých verzí ke dni 4.3.2013 podle toho, která zařízení přistoupila na *Google Play* v 14-ti denním rozmezí před výše zmíněným datem. Obrázek je převzat ze stránek Android Developers [4].

3.2 Architektura systému Android

Obsah této kapitoly (včetně obrázků) je čerpán především ze stránek Android Developers [6]. Architektura je rozdělena do 5-ti vrstev, které jsou zobrazeny na obrázku 3.2.

- **Aplikace** – Jedná se o nejvyšší vrstvu – jsou to aplikace, které používají uživatelé. Zařízení s operačním systémem Android se dodávají s řadou aplikací, jako je např. e-

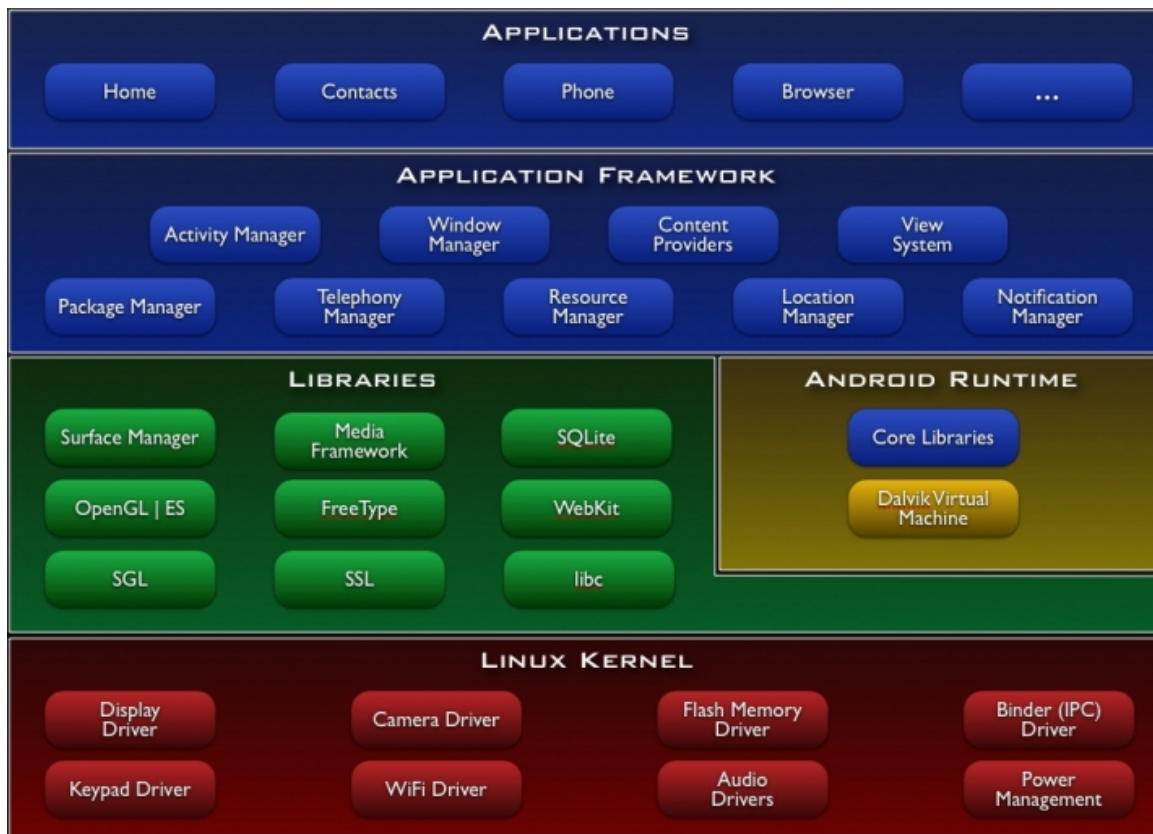


Obrázek 3.1: Verze Androidu

mailový klient, program na psaní SMS, mapy atd. Další aplikace je možné dodatečně nainstalovat např. z Google Play.

- **Aplikační framework** – Díky otevřené platformě poskytuje Android možnost budovat aplikace, které mají přístup ke stejnému API¹ jako používají systémové aplikace. Dovoluje přistupovat k hardwarovým zdrojům, informacím o lokaci, nastavovat alarmy a mnoho dalších funkcí. K důležitým částem patří:
 - **Views** – Komponenty pro tvorbu uživatelského rozhraní. Může se jednat o tlačítka, seznamy atd.
 - **Content providers** – Umožňuje aplikacím přijímat data od jiných aplikací a sdílet svá vlastní.
 - **Resource Manager** – Poskytuje přístup k externím zdrojům, které nejsou uloženy v kódu. Mohou to být textové řetězce, grafika nebo různé XML soubory.
 - **Notification Manager** – Aplikacím umožňuje zobrazovat upozornění ve stavovém řádku.
 - **Activity Manager** – Řídí životní cyklus aktivity a přepínání mezi nimi.
- **Knihovny** – Android poskytuje sadu C/C++ knihoven, které jsou využívány různými částmi systému. Knihovny jsou také poskytovány vývojářům skrze aplikační framework. Mezi hlavní knihovny patří:
 - **System C library** – Odvozená BSD systémová implementace standardní systémové C knihovny, upravená pro vestavěná Linuxová zařízení.
 - **Media Libraries** – Podporuje přehrávání, nahrávání a vytváření záznamů ve formátech MPEG4, H.264, MP3, AAC, AMR, JPG a PNG.
 - **3D libraries** – Knihovna pro vykreslování 3D grafiky založená na *OpenGL ES*.
 - **SQLite** – Odlehčený databázový systém pro aplikace.

¹rozhraní pro programování aplikací



Obrázek 3.2: Architektura

- **Android Runtime** – Vrstva sloužící převážně pro běh aplikací. Android obsahuje několik základních knihoven, které obsahují většinu funkcí dostupných v základních knihovnách jazyka Java. Každá aplikace běží ve svém vlastním procesu a s vlastní instancí *Dalvik* virtuálního stroje.

Dalvik je jednou z nejdůležitějších částí celého operačního systému Android. Je používán místo klasického virtuálního stroje od Javy, protože je optimalizován pro mobilní zařízení. Spouští soubory ve formátu *.dex*, který je optimalizovaný pro minimální paměťovou náročnost. Dalvik je založen na registrech a spouští třídy zkompilevané Java kompilátorem a převedené do formátu *.dex*.

- **Linuxové jádro** – Jedná se o nejnižší vrstvu. Android využívá jádro verze 2.6 pro základní systémové služby jako je bezpečnost, správa paměti, správa procesů, síťové prostředky a modely ovladačů. Slouží také jako abstraktní vrstva mezi hardwarem a zbytkem softwaru.

3.3 Verze systému

Od první oficiální verze (23.zář 2008) bylo již vydáno několik aktualizací, které jednak opravují nalezené chyby, ale také přidávají další funkce. Všechny verze jsou pojmenovány podle anglických názvů pro zákusky. Níže je uveden přehled a stručná charakteristika jednotlivých verzí, jak je uvedeno v [4].

- **Verze 1.0** – První oficiální verze. Oproti beta verzi značné vylepšení API (především přidán *launcher*), přidána galerie obrázků, nastavení a zajištění podpory pro *WIFI* a *bluetooth*.
- **Verze 1.1** – Představena 9.února 2009. Opět úprava API, možnost hlasového vytáčení, upraveno vyhledávání v mapových podkladech.
- **Verze 1.5, Cupcake** – Uvolněna 30.dubna 2009. Přidána možnost nahrávat a sledovat videa pořízená pomocí kamery, vylepšeno kopírování a možnost animovaného přechodu mezi jednotlivými obrazovkami. Další úpravy se týkaly především práce při psaní textových zpráv (automatické doplňování, vylepšená virtuální klávesnice).
- **Verze 1.6, Donut** – Vydána 15.zářím 2009. Především vylepšení *Android Marketu* – možnost rychlejšího vyhledávání. Optimalizováno hlasové ovládání telefonu a nově podpora pro rozlišení displeje *WVGA*.
- **Verze 2.0/2.1, Eclair** – Uvolněna 26.října 2009. Aktualizace uživatelského prostředí a Google map, podpora pro různé velikosti displeje a různá rozlišení. Nejvýznamnější změnou této verze je provedení optimalizace rychlosti hardwaru.
- **Verze 2.2.x, Froyo** – Aktualizace vydána 20.května 2010. Přidána podpora pro *Adobe Flash*, který ovšem není součástí systému, ale je možné jej stáhnout pomocí *Android Marketu*. Možnost instalace aplikací na paměťovou kartu, možnost vytvořit z telefonu *WIFI Hotspot*. Přidána podpora *OpenGL ES 2.0*.
- **Verze 2.3.x, Gingerbread** – Uvolněna 6.prosince 2010. Přidána podpora pro *SIP protokol* pro *IP telefonii*. Znovu vylepšení virtuální klávesnice a kopírování textu. Aktualizace Google map, již podpora 3D zobrazení. Vylepšení podpory pro vytváření aplikací s využitím *nativního* kódu.
- **Verze 3.x, Honeycomb** – Vydána 22.února 2011. První aktualizace pouze pro *tablety*. Přidána podpora pro multijádrové procesory, přidána stavová a "akční lišta". Virtuální klávesnici přidána možnost změny rozměrů.
- **Verze 4.0.x, Ice Cream Sandwich** – Aktualizace se objevila 19.října 2011. Vylepšena rychlost a výkon hardwaru. Virtuální ovládací tlačítka nahradila dříve používaná hardwarová. Vylepšena možnost odemykání zařízení za použití fotoaparátu, vylepšení e-mailového klienta, poskytovány jsou již i *offline* akce.
- **Verze 4.1.x, Jelly Bean** – Vydána 9.července 2012. Možnost libovolné úpravy a změny rozměrů u *widgetů*. *Android prohlížeč* webu nahrazen mobilní verzí *Google Chrome*. Přidána možnost *offline* hlasového ovládání.
- **Verze 4.2, Jelly Bean** – Zatím poslední verze, uvolněna 13.listopadu 2012. Přidán spořič obrazovky, zobrazující informace při nečinnosti. Na tabletech je možné mít více uživatelských účtů. Přidána aplikace hodiny s podporou světových časů, stopek a odpočítávání.

Z počátku byly jednotlivé verze vyvíjeny pouze pro mobilní telefony. Až když začal vývoj verze 2.3.x, začal současně také vývoj verze 3.0, která byla určena pro tablety. Následná verze 4.0 opět spojovala mobilní telefony a tablety.

3.4 Vývoj aplikací

Pro vývoj aplikací poskytuje Google Android soubor nástrojů nazývaný *Android SDK*, který je nutné doinstalovat do počítače. Podrobný návod lze nalézt na stránkách věnovaných se vývoji aplikací pro Android². Největší podpora je poskytována pro vývojové prostředí *Eclipse*, do kterého je nutné ovšem nainstalovat dodatečný *plugin ADT* podle návodu uvedeného na téže adrese. Obsah kapitoly je čerpán z knihy [2] a z [5]. Aplikace pro systém Android sestávají z následujících komponent:

- **Aktivita** – Jedná se vlastně o základní kámen pro tvorbu aplikací. Lze ji chápat jako analogii oken nebo dialogů aplikace pro stolní počítač. Jsou to samostatné části aplikace, které řeší funkční logiku celé aplikace. Životní cyklus aktivity je znázorněn na obrázku 3.3.

Aktivita se během svého životního cyklu může nacházet v některém z následujících stavů:

- **Aktivní** – Aktivita je spuštěna a běží na popředí.
- **Pozastavená** – Aktivita je spuštěna a je viditelná. Část obrazovky je ovšem překryta nějakým upozorněním. Aktivita je uživateli sice viditelná, ale nemůže s ní nijak interagovat. Tato situace může například nastat, když zařízení detekuje příchozí hovor a uživateli se zobrazí možnost jeho přijetí.
- **Zastavená** – Aktivita byla spuštěna, běží, ale je skryta za jinými aktivitami, které byly spuštěny po ní nebo se v jejím průběhu aktivovaly. Aktivita nemůže v tomto stavu uživateli přímo nic zobrazit, ale může s ním komunikovat prostřednictvím upozornění.
- **Mrtvá** – Aktivita buď nebyla vůbec spuštěna nebo došlo k jejímu násilnému ukončení.

Podle toho, v jakém stavu se aktivita nachází, volá některou z následujících metod:

- ***onCreate()*** – Tato metoda může být volána ve třech situacích:
 - * Při prvním spuštění aktivity (například po restartu systému). V tomto okamžiku se volá s parametrem `null`.
 - * Pokud již byla aktivita jednou spuštěna, volá se tato metoda s parametrem *Bundle*, který se získá z metody `onSaveInstanceState()`.
 - * Pokud již aktivita běžela a byly jí přiřazeny různé prostředky na základě různých stavů zařízení, lze aktivitu znovu vytvořit.
- ***onDestroy()*** – Tato metoda je volána při ukončení aktivity. A to buď při ukončení, kdy aktivita volá metodu `finish()` nebo pokud je aktivita ukončena násilně systémem.
- ***onStart()*** – Metoda je volána, když je aktivita poprvé spuštěna nebo proto, že byla opět aktivována po svém skrytí.
- ***onRestart()*** – Volání této metody probíhá v případě, že aktivita byla zastavena a restartuje se.
- ***onStop()*** – Volá se ve chvíli, kdy se má aktivita zastavit.

²<http://developer.android.com/sdk/installing/index.html>

- ***onPause()*** – Je volána v okamžiku, kdy se uživatel přepne mimo aktivitu a ta je tak pozastavena.
- ***onResume()*** – Volání této metody je výsledkem přesunutí aktivity do pozadí. Obvykle kvůli aktivaci jiné aktivity. V okamžiku volání této metody si systém Android vyhrazuje právo kdykoliv ukončit proces aktivity schované na pozadí.
- ***Broadcast receiver*** – Jedná se o systémové zprávy, které upozorňují aplikaci na výskyt různých událostí. Na tyto zprávy je možné reagovat nebo jejich prostřednictvím vytvářet a spouštět jiné aktivity, popřípadě je využívat k detekci specifických událostí.
- ***Content provider*** – Slouží k zajištění úrovně abstrakce jakýchkoli dat uložených v zařízení, která jsou přístupná různým aplikacím. Vývojový systém Android podporuje vývojáře v tom, aby zpřístupňovali svá data i jiným aplikacím než pouze své vlastní. Content provider poskytuje úplnou kontrolu nad způsobem přístupu k datům.
- ***Service*** – Na rozdíl od aktivit a Broadcast provideru jsou navrženy tak, aby (pokud je to třeba) pokračovaly ve své činnosti nezávisle na jakékoli aktivitě. Příkladem může být přehrávání hudby, které pokračuje i když už byla příslušná ovládací aktivita ukončena.

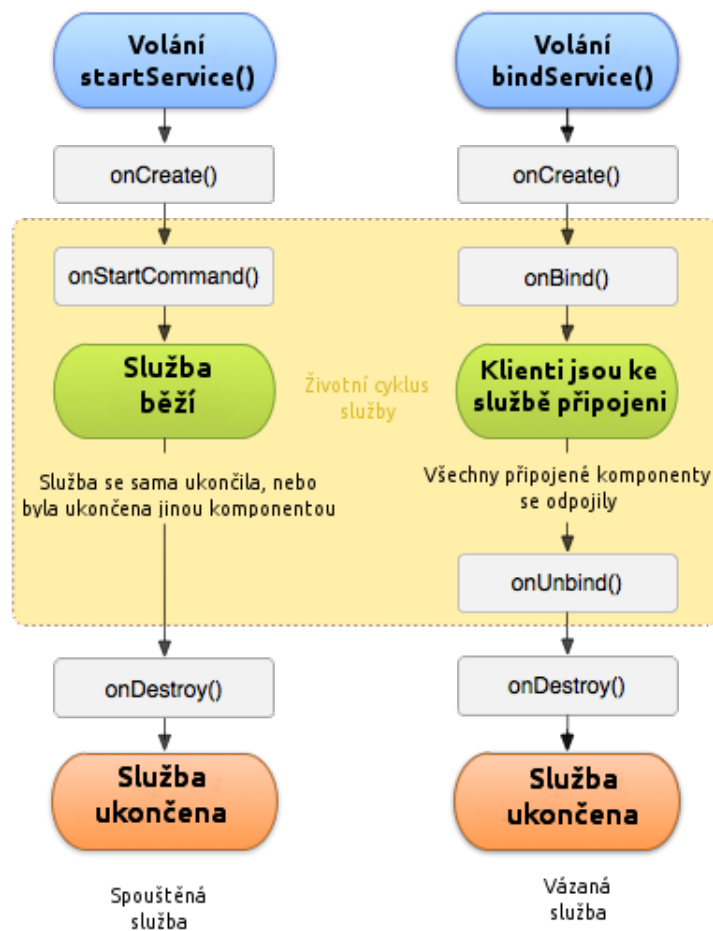
Existují dva druhy služeb:

- *Spouštěná služba* je spuštěna, když jiná komponenta aplikace volá metodu **`startService()`**. Pokud je služba spuštěna, může běžet na pozadí dokonce i v době, kdy už je ukončena komponenta, která ji spustila. Po dokončení úkolu se služba sama ukončí nebo může být ukončena jinou komponentou.
- *Vázaná služba* je spuštěna v případě, že se jiná komponenta naváže na službu voláním metody **`bindService()`**. Tato služba poskytuje komunikaci klient - server, která umožňuje komponentě komunikovat se službou, zasílat požadavky a přijímat odpovědi. K jedné službě může být připojeno více komponent a k ukončení služby dojde v případě, že se všechny komponenty odpojí.

V jedné službě lze ovšem používat oba výše zmíněné způsoby. Služba má svůj vlastní životní cyklus, který je zobrazen na obrázku [3.4](#).

Při vytvoření nového projektu pro platformu Android se vytvoří následující adresářová struktura:

- ***AndroidManifest.xml*** – Popisuje vyvíjenou aplikaci a komponenty, které aplikace obsahuje.
- ***build.xml*** – *Ant skript* pro kompilaci aplikace a její instalaci na zařízení.
- ***assets/*** – Statické soubory pro přibalení k aplikaci sloužící pro použití na zařízení.
- ***bin/*** – Složka pro uchování zkompilevané aplikace.
- ***gen/*** – Složka, do které překladač nástroje systému Android umístí zdrojový kód, který vygenerují.
- ***libs/*** – Složka uchovávající Java archivy třetích stran, které aplikace potřebuje.



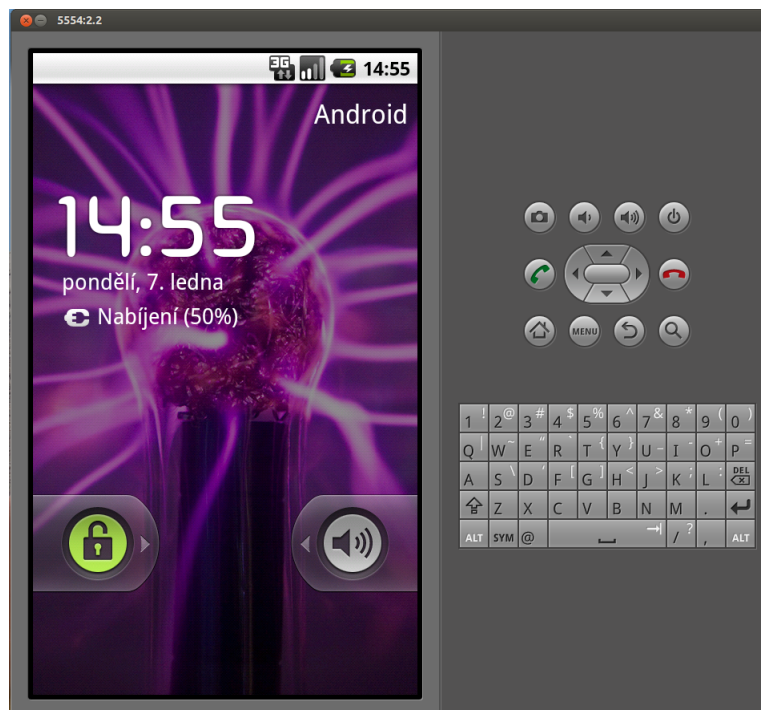
Obrázek 3.4: Životní cyklus služby

- *src/* – Složka, v níž jsou uloženy zdrojové kódy aplikace.
- *res/* – Složka uchovávající prostředky – například ikony nebo návrhy grafického rozhraní.
- *tests/* – Složka pro oddělený Android projekt, který je používán pro testování.

K otestování vyvíjené aplikace slouží *Emulátor* (znázorněn na obrázku 3.5), který je součástí Android SDK. Pomocí něj možné emulovat různá zařízení, ať už se liší například rozlišením obrazovky nebo třeba kvalitou hardwaru. Emulátor tedy poskytuje možnost testovat aplikace bez nutnosti vlastnit fyzické zařízení s OS Android. Testovat aplikace lze rovněž i na fyzických zařízeních, které je nutné propojit s PC pomocí USB kabelu.

3.5 Tvorba uživatelského rozhraní

V aplikacích pro Android je uživatelské prostředí postaveno na použití objektů *View* a *ViewGroup*. Je možné použít mnoho různých typů zobrazení, z nichž každé je ovšem potomkem třídy *View*.



Obrázek 3.5: Android emulátor

Třída `View` je základem pro všechny komponenty uživatelského rozhraní. Slouží také jako základní třída pro podtřídy nazývané *Widgety*, které slouží pro tvorbu interaktivních prvků uživatelského rozhraní, jako jsou např. tlačítka, textová pole atd.

Speciálním typem třídy je `ViewGroup`, jejímž obsahem jsou další třídy `View`. Jedná se o základní třídu pro podtřídy, kterým se říká rozvržení (*layout*), které poskytují možnosti pro různé druhy uspořádání architektury aplikace, jako je např. tabulkové, lineární, relativní, . . .

Existují dva přístupy, pomocí nichž lze vytvářet uživatelské rozhraní:

- **Deklarativní přístup** – Vzhled konkrétní obrazovky je popsán pomocí jazyka *XML*. Jak vypadá popis souboru ve formátu pdf je názorně vysvětleno v knize [1] od Pavla Herouta. Ve vývojovém prostředí Eclipse není nutné psát celý XML kód ručně, ale je možné si základní vzhled "naklikat" pomocí vestavěného editoru. Především kvůli omezení jazyka XML není tento přístup vhodný pro reakce na uživatelské vstupy.
- **Programovací přístup** – Slouží především pro tvorbu reakcí na uživatelské vstupy, jako je např. kliknutí na tlačítko, výběr textového pole atd.

Při vytváření skutečných aplikací se oba přístupy ve většině případů kombinují, kdy deklarativní se použije pro definici vzhledu a programovací pro reakce na uživatelské vstupy.

3.6 Lokace v systému Android

Chytré telefony jsou dnes schopny zjišťovat svoji polohu pomocí několika různých technologií, jako je *GPS*, *GSM* nebo *WIFI*. Tyto jednotlivé technologie poskytují různé příležitosti a omezení.

Obsah kapitoly je čerpán především z [7]. V zařízeních Android je možné získat aktuální pozici díky systémové službě, jejímž základním prvkem je třída `LocationManager`. Stejně jako u ostatních systémových služeb nelze instanci této třídy získat přímo, ale je třeba požádat o instanci systém.

Získání aktuální pozice funguje pomocí zpětného volání. Zažádá se o získání aktuální pozice od `LocationManager` voláním `requestLocationUpdates()`. Při volání je zvolen poskytovatel polohy, minimální čas a vzdálenost, po kterých je poskytována další informace o poloze a poskytovatel informací. Poskytovatelem je nejčastěji instance rozhraní `LocationListener`. `LocationListener` musí obsahovat několik zpětných metod, které `LocationManager` volá v případě, že se změní pozice uživatele nebo při změně stavu služby. Mezi tyto metody patří:

- `onLocationChanged()` – Metoda je volána v případě změny polohy.
- `onStatusChanged()` – Volání metody v případě změny dostupnosti polohy.
- `onProviderEnabled()` – Tato metoda je volána, pokud je uživatelem povolen poskytovatel polohy.
- `onProviderDisabled()` – Dochází k volání metody v případě zakázání poskytovatele polohy uživatelem.

K získání informací o poskytovateli polohy slouží abstraktní třída `LocationProvider`. Je možné vybrat mezi dvěma poskytovateli polohy – jejich popisu se věnuje následující část této podkapitoly.

- **GPS** – *Global Positioning System* (GPS) je vojenský globální družicový systém provozovaný Ministerstvem obrany Spojených států amerických, který poskytuje celosvětové pokrytí (vyjma míst v budovách nebo podzemí). Tento systém je schopen určit polohu kdekoli na zemi nebo nad zemí s odchylkou do 10-ti metrů. Pomocí GPS je také možné určovat přesný čas. Nevýhodou použití GPS na zařízeních s operačním systémem Android je velká spotřeba baterie.

Systém GPS lze rozdělit do tří podsystémů:

- **Kosmický** – Systém tvoří 24 družic, které obíhají kolem zemského povrchu rychlostí 3,8 km/s ve výšce asi 20000 km po 6-ti oběžných dráhách. Každá jednotlivá družice obsahuje přijímač, vysílač, atomové hodiny a další přístroje.
 - **Řídící(kontrolní)** – Slouží k monitorování kosmického podsystému, získané údaje zasílá zpět družicím a řídí jejich pohyb a správný chod atomových hodin.
 - **Uživatelský** – Díky GPS přijímačům mohou uživatelé získávat data z družic, a to např. přesnou polohu, nadmořskou výšku nebo přesný čas.
- **WIFI, GSM** – Technologie založené na WIFI³ a GSM⁴ mohou potencionálně poskytovat relativně hrubý odhad všude tam, kde je WIFI, případně mobilní signál dostupný, ovšem je třeba připojení k internetu. Přesnost je oproti GPS menší, navíc je tato technologie závislá na množství GSM vysílačů a kvalitě GSM signálu, ovšem oproti GPS šetří baterii.

³označení pro standardy *IEEE 802.11* popisující bezdrátovou komunikaci v počítačových sítích

⁴nejrozšířenější standard pro mobilní telefony

3.7 Google Maps API

Pomocí Google Maps API lze do aplikací vyvíjených na operační systém Android přidávat mapy založené na *Google Maps*. Tato externí knihovna (poskytovaná Googlem) umožňuje automatický přístup k serveru Google Maps, stahování, zobrazování a práci s mapami.

Při vývoji aplikací je nutné tuto knihovnu dodatečně přidat do Android SDK, jelikož standardně v něm není obsažena. Aby bylo možné knihovnu používat, je tvůrce aplikace nucen získat klíč (*Maps API Key*), který je registrován na jednoho určitého vývojáře. K registraci dochází prostřednictvím vývojářova Google účtu. Popis získání a zaregistrování klíče je uveden na Android Developers⁵.

API rovněž poskytuje možnost přidávat značky, polygony a překrývat základní mapu, aby bylo možné změnit uživatelské zobrazení konkrétní mapy. Tyto objekty poskytují dodatečné informace o mapě a poskytují také možnost interakce uživatele s mapou.

Knihovna obsahuje speciální třídu – *MapView*, která slouží k zobrazování mapy. Tato třída umožňuje pracovat s mapami stejným způsobem jako s ostatními prvky view, zmíněnými v kapitole 3.5

⁵<http://developer.android.com/google/play-services/maps.html>

Kapitola 4

Návrh aplikace

Tato kapitola se zabývá návrhem aplikace podle uživatelských požadavků a důvody, proč by měla aplikace vůbec vzniknout.

Vytvořit aplikaci pro tvorbu poznámek jsem se rozhodl především z důvodu, že jsem relativně dlouho hledal aplikaci, která by mi vyhovovala a splňovala všechny mé požadavky. Našel jsem několik aplikací, které fungovaly téměř bezchybně, viz kapitola 2, nicméně každé z nich nějaká mnou požadovaná funkčnost scházela nebo mi příliš nevyhovovala.

4.1 Uživatelské požadavky

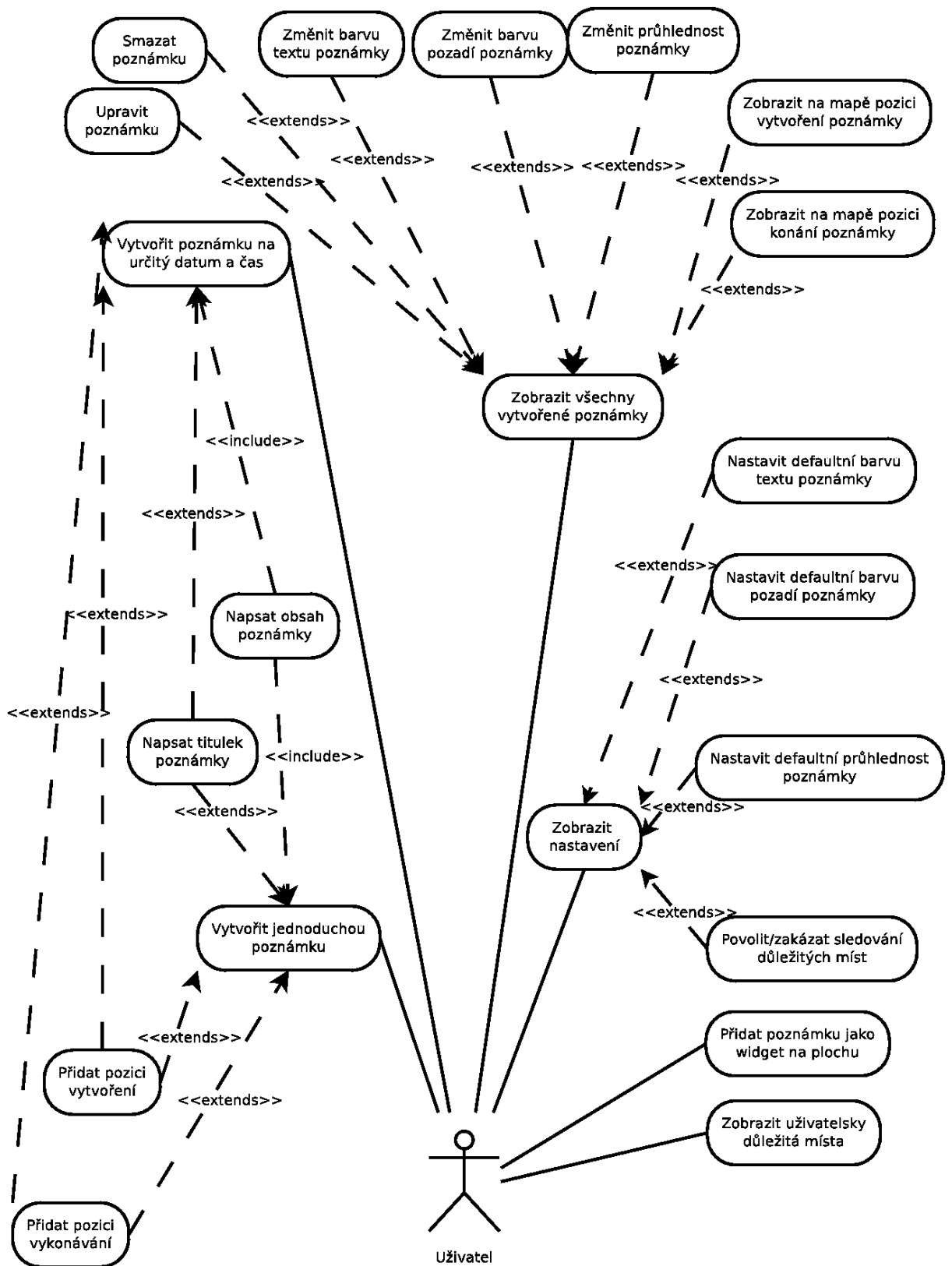
Aplikace bude uživatelům poskytovat možnost přidat jednoduchou textovou poznámku nebo poznámku s upozorněním na určité datum a čas. Každá poznámka obsahuje nadpis, který je volitelný a obsah poznámky, který je povinný. Při vytváření bude mít uživatel možnost přidat k poznámce pozici, kde byla vytvořena a místo konání poznámky (např. v případě schůzky). Pozice vytvoření poznámky bude získána pomocí satelitů GPS nebo pomocí bezdrátových sítí. Místo konání poznámky bude uživatel moci přidat manuálně – buď kliknutím na mapu nebo zadáním přesných souřadnic. Vytvořené poznámky bude uživatel moci vkládat na plochu jako widgety a měnit styl jejich zobrazení. Dále bude možné zobrazovat na mapě místa, kde byly poznámky vytvořeny, případně místa jejich konání.

Při každém spuštění aplikace (pokud to uživatel nezakáže v nastavení) se rovněž spustí proces na detekci sémanticky důležitých míst pro uživatele. Automaticky se bude zaznamenávat pohyb uživatele a na základě shlukování těchto pozic bude možné určit místa, kde se uživatel vyskytuje nejčastěji a tráví zde většinu svého času.

Pokud bude aplikace spuštěna a uživatel bude mít vytvořenu nějakou poznámku s místem vykonávání, bude se kontrolovat aktuální pozice a v případě, že se uživatel přiblíží k místu vykonávání, bude mu nabídnuta možnost zobrazení této konkrétní poznámky.

Data o aktuální pozici potřebná k odhadu uživatelsky důležitých míst budou sbírána při každém spuštění aplikace (pokud to uživatel implicitně nezakáže v nastavení) a budou ukládána pro další zpracování.

Na obrázku 4.1 je zobrazen *Use Case* diagram, kde jsou naznačeny možnosti uživatele.



Obrázek 4.1: Use Case diagram

Kapitola 5

Implementace aplikace

Tato kapitola popisuje implementaci aplikace Pokročilé poznámky pro Android podle vědomostí získaných v předcházejících kapitolách. Pro tvorbu aplikace bylo využito vývojové prostředí Eclipse s doinstalovaným modulem ADT, který obsahoval Android SDK. Veškeré zdrojové kódy jsou umístěny v balíku `com.fit.vutbr.AdvancedNote`.

V následujících kapitolách jsou popsány jednotlivé části aplikace, jejich funkčnost a uživatelské rozhraní.

5.1 Části aplikace

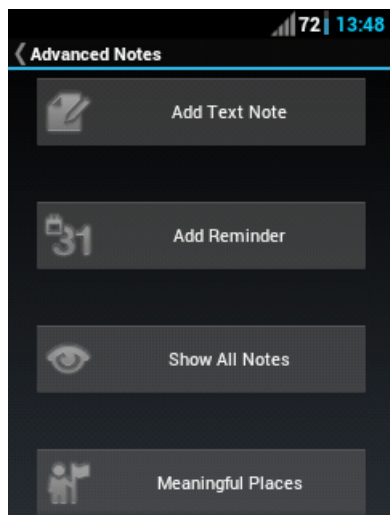
- **Domovská obrazovka** – Jedná se o aktivitu, která se zobrazí při prvotním spuštění aplikace. Aktivita obsahuje sadu tlačítek pro možnost přechodu do dalších aktivit. Je možné přejít k přidání jednoduché textové poznámky, přidání poznámky s upozorněním, zobrazení seznamu všech poznámek. Pokud má uživatel povolenu detekci sémanticky důležitých míst, nachází se zde ještě tlačítko pro možnost zobrazení těchto míst na mapě. Rovněž je možné v této aktivitě vyvolat menu, ze kterého je možné zobrazit nastavení aplikace, zobrazit informace o aplikaci a také aplikaci korektně ukončit. Implementace je umístěna v souboru `AdvancedNotes.java`.

Obrázek 5.1 ukazuje, jak Domovská obrazovka vypadá.

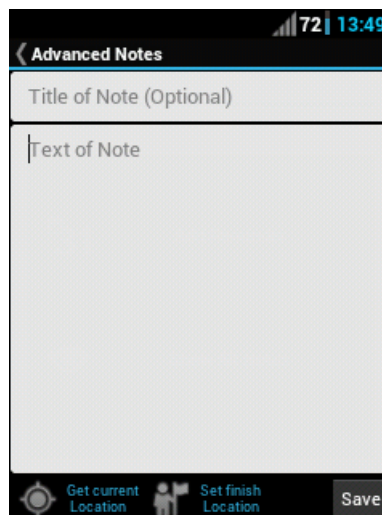
- **Přidání jednoduché poznámky** – Aktivita, která umožňuje přidat do databáze jednoduchou textovou poznámku. Aby bylo možné poznámku uložit, je nutné vyplnit text poznámky. Mezi volitelné možnosti patří nadpis poznámky, místo vytvoření a místo konání poznámky. Získání aktuální pozice je možné pomocí satelitů GPS nebo pomocí bezdrátových sítí – záleží na tom, které určení polohy má uživatel povoleno v nastavení systému. V případě, že nemá povolenu ani jednu možnost pro získání aktuální pozice a přesto se pokusí aktuální pozici získat, je mu nabídnuta možnost přejít do nastavení systému a alespoň jednu z možností povolit. Přidání místa konání poznámky je možné buď zadáním přesných souřadnic nebo kliknutím na mapu (záleží na tom, kterou možnost má uživatel vybranou v nastavení). Po potvrzení je poznámka uložena do databáze a uživatel se přesouvá opět na domovskou obrazovku. Detailní popis databáze lze nalézt v kapitole 5.4.

Jak vypadá aktivita pro přidání jednoduché poznámky je ukázáno na obrázku 5.2

- **Přidání poznámky s upozorněním** – Tato aktivita je totožná s aktivitou pro přidání jednoduché poznámky, jen je rozšířena o volbu data a času, kdy má být na



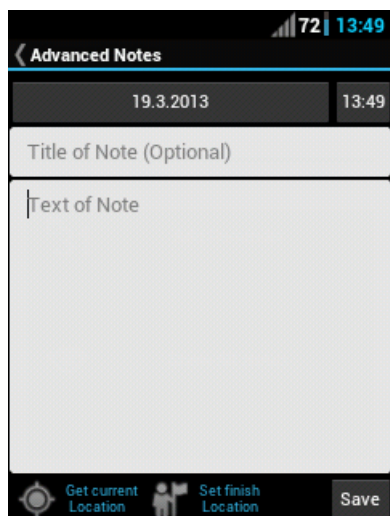
Obrázek 5.1: Úvodní obrazovka



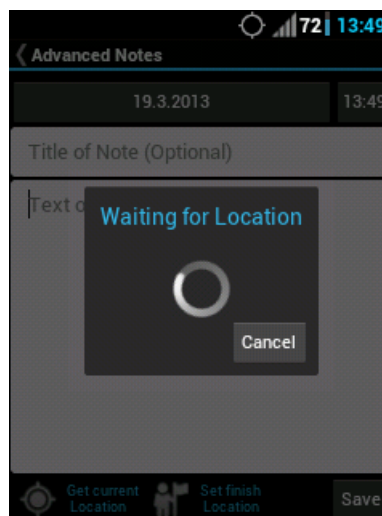
Obrázek 5.2: Textová poznámka

poznámku upozorněno. Implementaci přidávání obou typů poznámek a stejně tak jejich editaci lze nalézt v souboru `AddNote.java`.

Na obrázku 5.3 je ukázáno, jak vypadá přidávání poznámky s upozorněním a na obrázku 5.4 lze vidět čekání na získání aktuální pozice při vytváření poznámky.



Obrázek 5.3: Poznámka s upozorněním



Obrázek 5.4: Čekání na aktuální pozici

- **Seznam všech poznámek** – Aktivita, která zobrazí seznam všech poznámek uložených v databázi podle data vytvoření, případně úpravy. Jednotlivé poznámky již mohou být označeny jako ukončené (vykonané) – to, zda jsou v seznamu také zobrazeny záleží na uživatelském nastavení. Pokud jsou zobrazeny, pak je jejich text přeškrtnut, aby je bylo možné vizuálně oddělit od ostatních poznámek. U každé poznámky jsou zobrazeny následující věci:

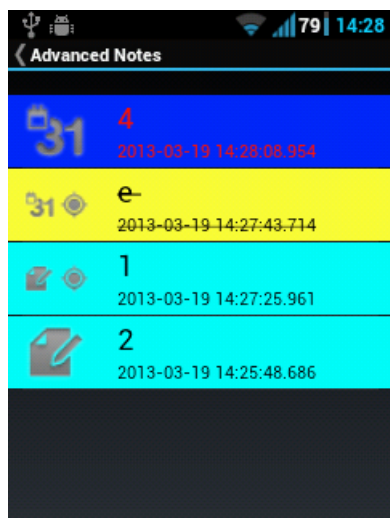
- Nadpis poznámky (pokud byl vyplněn) nebo oříznutý začátek textu poznámky.
- Datum vytvoření, případně úpravy poznámky.
- Ikona, která značí typ poznámky.
- Ikona značící, zda je u poznámky přiřazeno místo vytvoření, případně místo konání a je možné jej zobrazit pomocí Google map.

Při kliknutí na poznámku se spustí aktivita, která umožňuje editaci poznámky.

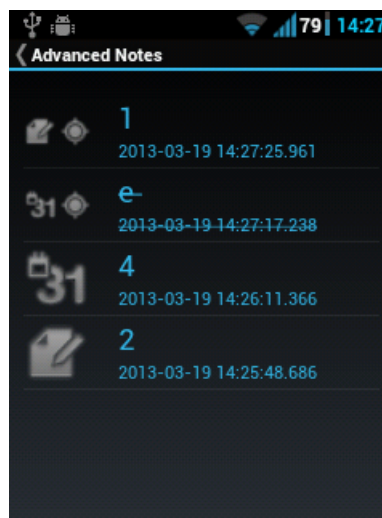
U každé poznámky je rovněž možné vyvolat kontextové menu s následujícími položkami:

- Smazat poznámku.
- Nastavit barvu pozadí poznámky.
- Nastavit průhlednost poznámky.
- Nastavit barvu textu poznámky.
- Označit jako ukončenou.
- Pokud je u poznámky místo vytvoření nebo vykonávání, zobrazit ji na mapě.

Poznámky mohou být barevně odlišeny – obrázek 5.5 nebo mohou být zobrazeny s defaultním stylem aplikace – obrázek 5.6. Styl zobrazení záleží na uživateli a na jeho nastavení. Vyvolané kontextové menu pro poznámku je ukázáno na obrázku 5.10. Tato aktivita je implementována v souboru `AllNotes.java`.



Obrázek 5.5: Seznam poznámek s barevným odlišením



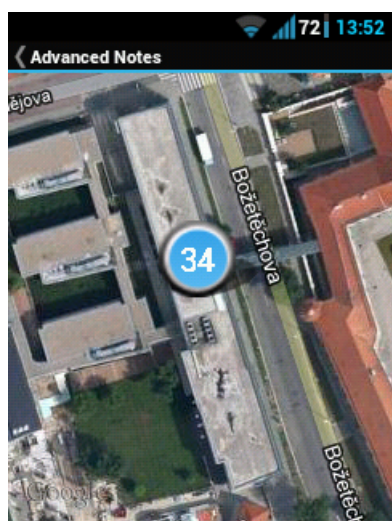
Obrázek 5.6: Seznam poznámek s defaultním stylem

- **Zobrazení uživatelsky důležitých míst** – Tato mapová aktivita zobrazí uživatelsky důležitá místa. Aby nebylo na mapě vykreslováno příliš mnoho značek a nedocházelo tak k situaci, kdy se mapa stane pro uživatele nepřehlednou, jsou místa vyskytující se blízko sebe shlukována do jedné značky, uvnitř které je vypsán počet výskytů v tomto okolí.

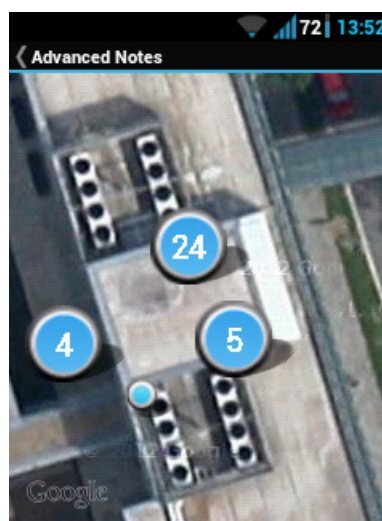
Shlukování značek do jedné je implementováno následujícím způsobem: Jelikož je známa velikost značky, je tedy jasné v jaké minimální vzdálenosti od sebe musí být

značky umístěny, aby nedocházelo k jejich překrytí. Skutečná vzdálenost mezi body na mapě je tedy přepočtena na vzdálenost na obrazovce v pixelech. Pokud je vzdálenost mezi dvěma samostatnými body menší než velikost značky, jsou tyto dva body spojeny do jednoho a dovnitř značky je vepsán počet bodů. Takto jsou procházeny všechny body a je vytvořena výsledná podoba mapy. Při každém přiblížení nebo oddálení pohledu na mapu je vzhled aktualizován.

Na obrázku 5.7 je zobrazena oddálená mapa – body jsou tedy spojeny do jednoho, aby nedocházelo k jejich překryvu. Naproti tomu na obrázku 5.8 je mapa přiblížena – některé body jsou již tedy zobrazeny samostatně, jelikož je pro ně na displeji dostatek místa.



Obrázek 5.7: Shluknuté body



Obrázek 5.8: Částečně samostatné body

- **Nastavení** – Jedná se o tzv. *PreferenceActivity*, viz [8], ve které je možné provádět uživatelské nastavení. Ukládání dat probíhá pomocí *Shared Preferences*, jejichž detailní popis lze nalézt na stránkách Android Developers[9].

Celé nastavení je rozděleno na dvě podseky, konkrétně na globální nastavení a nastavení vzhledu poznámky. V globální části lze nastavit následující věci:

- **Povolit/zakázat barevné poznámky** – Pokud je tato možnost povolena, pak se uživateli v seznamu poznámek zobrazují jednotlivé poznámky s příslušnou barvou textu a barvou pozadí podle toho, jaké barvy uživatel zvolí.
- **Povolit/zakázat získávání sémanticky důležitých míst** – Pokud je povolena tato možnost, pak je při spuštění aplikace spuštěn proces pro vyhledávání sémanticky důležitých míst. Detailní popis procesu lze nalézt v kapitole 5.2
- **Povolit/zakázat automatické otevření poznámky** – V případě povolení je možné automaticky otevírat poznámky tehdy, pokud se uživatel blíží k místu konání. V závislosti na této položce (pokud je povolena) má uživatel možnost nastavit vzdálenost od místa konání, kdy má být poznámka otevřena. Podrobnosti jsou uvedeny v kapitole 5.3.
- **Zvolit styl zobrazení mapy** – Dovoluje uživateli vybrat ze dvou stylů zobrazení mapy, konkrétně satelitní nebo klasické.

- **Zvolit styl zadávání místa konání** – Umožňuje uživateli vybrat, zda bude místo konání zadáváno pomocí přesných souřadnic nebo kliknutím na mapu.
- **Povolit/zakázat zobrazení ukončených poznámek** – Pokud je položka povolena, pak se v seznamu všech poznámek zobrazují i již ukončené poznámky (od neukončených jsou odlišeny tím, že je jejich text přeškrtnut).

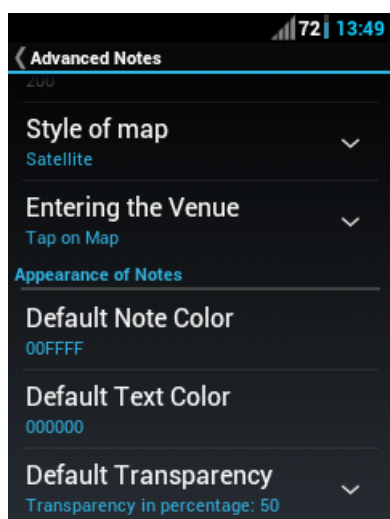
V části pro vzhled poznámky lze nastavit tyto věci:

- **Barvu pozadí poznámky** – Barva pozadí ovlivňuje jednak vzhled poznámky v seznamu poznámek, tak i (a to především) vzhled widgetu na domovské obrazovce.
- **Barvu textu poznámky** – Určuje barvu textu poznámky.
- **Průhlednost poznámky** – Nastavení průhlednosti widgetu v případě, že je umístěn na domovské obrazovce.

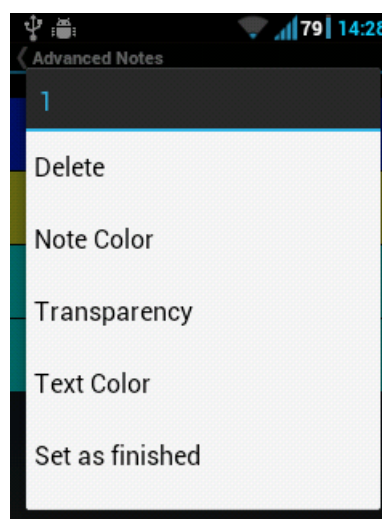
Implementace se nachází v souboru `Setting.java`.

Výběr barvy je implementován v souboru `MyColorPicker.java`, který je převzat z příkladů dodávaných spolu s Android SDK.

Část pro nastavení vzhledu poznámky je zobrazena na obrázku 5.9



Obrázek 5.9: Nastavení



Obrázek 5.10: Nastavení pro konkrétní poznámku

- **Widgety** – Aplikace umožňuje na domovskou obrazovku vkládat dva typy widgetů – malé widgety obsahující konkrétní poznámku s příslušným nastaveným vzhledem a větší widget obsahující seznam aktuálních poznámek a možnost rychlého přístupu do nastavení a vytváření poznámek.

5.2 Získávání uživatelsky důležitých míst

Při každém zapnutí aplikace (pokud to uživatel nezakáže v nastavení) je automaticky spuštěn proces, který se stará o získávání aktuální pozice. Proces je spuštěn jako služba – podrobnější popis je uveden v kapitole 3.4. Sběr dat probíhá z několika zdrojů. Primárně

se aplikace snaží získat aktuální pozici pomocí GPS satelitů, jelikož takto získaná pozice je nejpřesnější. Současně sběr dat probíhá pomocí bezdrátových sítí, jelikož GPS pozice nelze zpravidla získat uvnitř budov, zatímco pomocí bezdrátových sítí je to možné, pokud je dostupné připojení k internetu. Snaha o získání aktuální pozice je automaticky spuštěna každou minutu i v okamžiku, kdy není aktivita viditelná. V momentě, kdy je získána aktuální pozice, je ihned uložena do databáze pro další použití.

V případě, že není možné získat aktuální pozici ani pomocí GPS satelitu, ani pomocí bezdrátových sítí, pak je spuštěn poslední pokus o získání pozice. V případě, že je dostupný GSM signál, je možné získat aktuální pozici tímto způsobem. V tomto případě je nutné získat identifikační číslo vysílače (CID), ze kterého je do zařízení přijímán signál a kód území, na kterém se vysílač nachází (LAC). Pokud je dostupné připojení z internetu, je okamžitě pomocí dostupného *API*¹ stažena přesná pozice vysílače, která je uložena do databáze. V případě nedostupného připojení k internetu jsou CID a LAC uloženy rovněž do databáze a v momentě, kdy je připojení dostupné, jsou opět pomocí *API* převedeny na přesnou polohu vysílače. Určení polohy jen podle pozice vysílače není příliš přesné, ale v krajním případě se dá použít alespoň pro přibližný odhad místa.

Rovněž v případě, kdy uživatel získá svou aktuální pozici při vytváření poznámky a má v nastavení povoleno sledování sémanticky důležitých míst, je tato pozice uložena do databáze pro pozdější zpracování. Implementace se nachází v souboru `ImportantPlaces.java`.

5.3 Automatické otevření poznámky

Pokud má uživatel v nastavení tuto možnost povolenou, je opět každou minutu provedena snaha o získání aktuální pozice, jak je popsána v kapitole 5.2. Pro všechny poznámky, které mají vyplněno místo konání, je vypočtena vzdálenost mezi aktuálním místem a místem konání. Pokud je tato vzdálenost menší nebo rovna vzdálenosti vyplněné v nastavení, je uživateli zobrazeno okno se seznamem těchto poznámek a je mu umožněno jejich rychlé zobrazení.

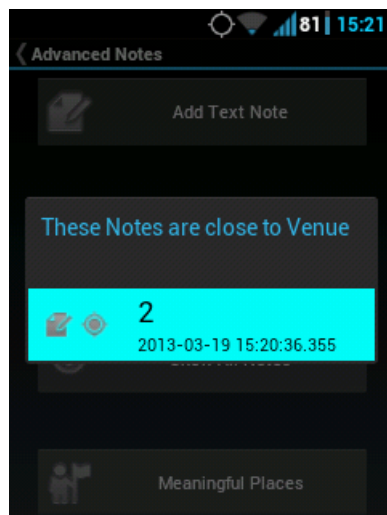
Jak vypadá okno se seznamem poznámek je ukázáno na obrázku 5.11

5.4 Ukládání do databáze

Aplikace při práci s poznámkami a widgety využívá vestavěnou *SQLite databázi*. Podrobné informace o práci s databází lze získat ze stránek Android Developers[10]. Vyvíjená aplikace obsahuje celkem čtyři databázové tabulky. Jejich podrobnější popis a strukturu přikládám níže.

- Tabulka **notes** – Jedná se o stěžejní databázovou tabulku celé aplikace, která slouží k ukládání poznámek obou výše zmíněných typů. Tabulka obsahuje následující položky:
 - **id** – Identifikační číslo poznámky.
 - **title** – Nadpis poznámky. Může být prázdný, pokud jej uživatel při vytváření poznámky nevyplnil.
 - **note** – Obsah poznámky. Vždy musí být vyplněn. Bez něj nepovolí aplikace uživateli vložit poznámku do databáze.

¹<http://www.google.com/glm/mmap>



Obrázek 5.11: Okno se seznamem blízkých poznámek

- **latitude** – Zeměpisná šířka. Je vyplněna v případě, kdy uživatel zjistil svoji aktuální pozici při vytváření poznámky.
 - **longitude** – Zeměpisná délka. Platí pro ni stejná pravidla jako pro zeměpisnou šířku.
 - **timestap** – Časová značka. Jedná se o datum a čas, kdy byla poznámka vytvořena (případně naposledy upravována).
 - **color** – Barva pozadí poznámky. Pokud ji uživatel manuálně nezmění, je brána defaultní barva pozadí z nastavení.
 - **text_color** – Barva textu poznámky. Platí pro ni stejná pravidla jako pro barvy pozadí.
 - **transparency** – Průhlednost poznámky. Pokud průhlednost uživatel sám nezmění, je defaultní barva brána dle nastavení.
 - **date** – Datum připomenutí. Určuje den, kdy má být na poznámku upozorněno.
 - **time** – Čas připomenutí. Určuje hodinu připomenutí. V případě, že jsou položky datum a čas vyplněny, jedná se o poznámku s upozorněním.
 - **latitude_finish** – Zeměpisná šířka cíle. Je vyplněna v případě, kdy uživatel při vytváření nebo úpravě poznámky zadal místo konání.
 - **longitude_finish** – Zeměpisná délka cíle. Má stejné požadavky jako zeměpisná šířka cíle.
 - **state** – Stav poznámky. Určuje, zda je poznámka aktuální nebo již označena jako ukončená.
- Tabulka **places** – Tabulka sloužící k ukládání sémanticky důležitých míst. Obsahuje následující položky:
 - **id** – Identifikační číslo.
 - **longitude** – Zeměpisná délka místa.

- **latitude** – Zeměpisná šířka místa.
- Tabulka **offline_places** – Je užívána k ukládání CID a LAC zmíněných v kapitole 5.2, kdy není dostupné připojení k internetu a není tak možné získat skutečnou pozici vysílače. Tabulka obsahuje následující sloupce:
 - **id** – Identifikační číslo.
 - **cid** – Identifikační číslo vysílače.
 - **lac** – Kód území, na kterém se vysílač nachází.
- Tabulka **widget** – Tato tabulka je používána v případě, kdy uživatel vkládá na obrazovku jednotlivé poznámky jako widgety. Operační systém Android přiděluje každému widgetu, který je umístěn na domovskou obrazovku, unikátní identifikační číslo. Toto číslo je uloženo do tabulky společně s identifikačním číslem poznámky. V okamžiku, kdy se uživatel snaží upravit text poznámky kliknutím na widget, aplikace si zjistí identifikační číslo widgetu a z tabulky si získá identifikační číslo poznámky – je tedy jasné, jaká poznámka má být upravována. Tabulka obsahuje tyto položky:
 - **id** – Identifikační číslo.
 - **widget_id** – Identifikační číslo widgetu přidělené operačním systémem.
 - **note_id** – Identifikační číslo poznámky z tabulky notes.

Veškerá práce s databází je prováděna v souboru `MySqliteDatabase.java`.

Kapitola 6

Testování

Nedílnou součástí při vývoji aplikace je i její testování. V této kapitole je popsáno na jakých strojích testování probíhalo a jaké chyby (případně nedostatky) byly zjištěny. Celá aplikace byla vyvíjena v operačním systému *Ubuntu 12.10 Quantal Quetzal* ve vývojovém prostředí *Eclipse 3.8.0* s doinstalovaným *ADT pluginem*.

6.1 Testovaná zařízení

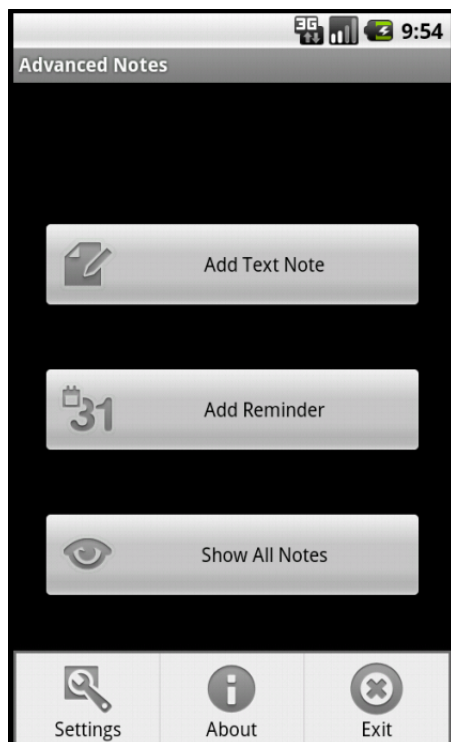
Při vývoji byla aplikace nejčastěji testována na emulátoru, na kterém běžel operační systém Android verze 2.2 s rozlišením displeje 320x480. Na emulátoru docházelo především k testování uživatelského vzhledu aplikace. Během celého vývoje byla aplikace rovněž testována na zařízení *HTC Wildfire*, jehož operační systém byl verze 2.3.7 a rozlišení displeje rovněž 320x480. Testování na tomto reálném zařízení probíhalo v době, kdy bylo vyvíjeno získávání aktuální pozice, jelikož občas je k získání pozice nutné použít GSM signál, který není možné v emulátoru nijak emulovat.

V době, kdy byla aplikace podle požadavků plně funkční na obou výše zmíněných zařízeních, byla následně také otestována na následujících zařízeních:

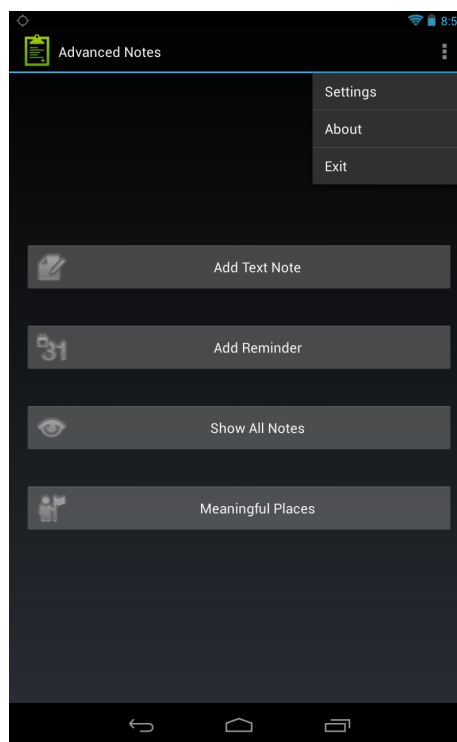
- **Sony Xperia Ray** – verze 4.0.4
- **Sony Xperia Sola** – verze 4.1.2
- **Vodafone 945** – verze 2.2
- **Samsung Galaxy S2** – Verze 4.0.1
- **Asus Nexus7** – verze 4.2.2

Aplikace na všech testovaných zařízeních fungovala korektně. Pouze na tabletu Nexus7 nebylo možné vyvolat menu na domovské stránce, jelikož na tomto zařízení již není tlačítko pro vyvolání menu, ale je zde možnost zobrazit tlačítko v tzv. akční liště. Aplikace byla tedy optimalizovaná následně – jelikož telefon pozná, jaká verze operačního systému je k dispozici, zvolí dle toho vhodný styl aplikace. Jak vypadá vyvolané menu je pro porovnání ukázáno na obrázcích [6.1](#) a [6.2](#).

Dalším drobným nedostatkem na zařízeních Samsung Galaxy S2 a Nexus7 bylo, že dialog pro výběr barvy se na displeji zobrazoval velmi malý a práce s ním byla nepohodlná. Došlo tedy k této optimalizaci – před vytvářením dialogu je zjištěna velikost displeje a dle tohoto kritéria je nastavena velikost dialogu.



Obrázek 6.1: Vyvolané menu ve verzi 2.2

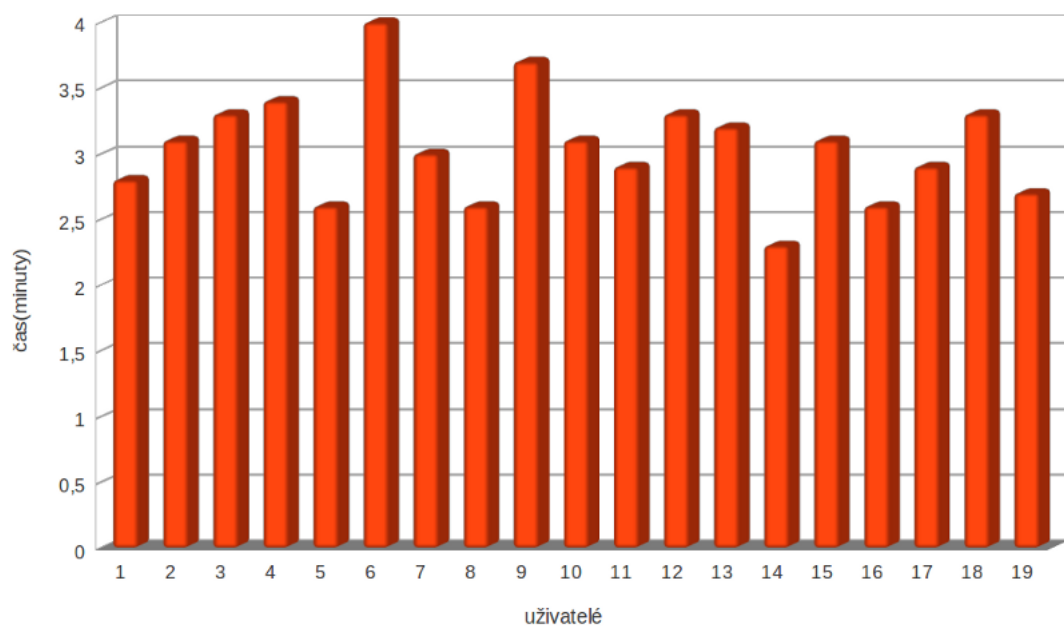


Obrázek 6.2: Vyvolané menu ve verzi 4.2.2

6.2 Testování uživateli

Po dokončení a odladění aplikace došlo k otestování funkčnosti aplikace pomocí uživatelů. Záměrně byli pro testy vybráni jak uživatelé, kteří se v operačním systému Android vyznají, tak i uživatelé, kteří s ním nemají příliš velké zkušenosti. Všichni měli čas 5 minut na seznámení se s aplikací, poté došlo k vlastnímu testování. Uživatelům byla zadána sada úkolů (viz příloha), které měli splnit a byl jim měřen čas. Výsledky jsou vyneseny v grafu 6.3.

Doba plnění úkolů v mém případě byla 2 minuty 30 sekund. Z grafu je tedy patrné, že žádný z uživatelů se pod tuto hranici nedostal. Ovšem někteří se jí velmi přiblížili. Průměrný čas plnění úkolů je 3 minuty 10 sekund. Lze tedy vyvodit, že aplikace není pro uživatele příliš náročná na ovládání, což bylo jedním z hlavních cílů kladených na její implementaci.



Obrázek 6.3: Graf výsledků testování

Kapitola 7

Závěr

Cílem této bakalářské práce bylo seznámit se s principy operačního systému Android a tvorbou aplikací pro tuto platformu. V rámci práce jsou podrobněji popsány již existující aplikace, zabývající se správou poznámek – jejich možnosti a především jejich nedostatky. Součástí práce je také návrh aplikace pro správu poznámek na operačním systému Android s využitím senzorů mobilní platformy. Poslední část práce se věnuje implementaci mnou navržené aplikace a zároveň jejímu důslednému otestování.

Největším přínosem práce pro mě bylo seznámení se s principy platformy Android. Naučil jsem se navrhovat a vytvářet aplikace pro tuto, v dnešní době nejrozšířenější platformu. Díky této bakalářské práci jsem se lépe seznámil s programovacím jazykem Java a naučil jsem se vytvářet soubory ve formátu XML. Naučil jsem se pracovat s SQLite databází, která je využívána ve většině aplikací vyvíjených pro operační systém Android. V neposlední řadě jsem si osvojil práci s Android SDK, který je využíván pro vývoj aplikací a testování na různých verzích systému.

Dle mého názoru a výsledků uvedených v této bakalářské práci se domnívám, že aplikace by v dalším vývoji mohla být vylepšena hned v několika oblastech. Dala by se přidělat možnost vytvářet další typy poznámek, jako jsou např. zvukové, obrázkové nebo mluvené. Aplikace by také mohla být lépe optimalizována pro tablety – v době, kdy uživatel prochází seznam poznámek a přejde do detailu jedné konkrétní, neotevřela by se tato poznámka na celou stránku. Detail poznámky by zůstal pouze na polovině této stránky, na druhé by byl k dispozici seznam poznámek. Vzhledem k velikosti displeje u tabletů by tato optimalizace uživatelům zcela jistě zpříjemnila práci. Další možností rozšíření by bylo sesynchronizovat aplikaci např. s Google kalendářem, aby se zamezilo ztrátě poznámek. Dalším, poměrně pokročilým rozšířením, by vzhledem k možnosti zadávání místa konání u poznámek mohla být také možnost navigovat uživatele právě k tomuto místu. Do budoucna by se také dalo upravit uživatelské rozhraní aplikace, které je v současné době postaveno pouze na základním vzhledu prvků, které Android poskytuje.

Literatura

- [1] Herout P.: *Java a XML pro Javu 5 i 6*. KOOP, 2007, iISBN 80-7232-307-5.
- [2] Murphy M. L.: *Android 2: Průvodce programováním mobilních aplikací*. COMPUTER PRESS, 2011, iISBN 9788025131947.
- [3] Syed H.: The History of Android: From Cupcakes to Jelly Beans. <http://droidlessons.com/the-history-of-android-from-cupcakes-to-jelly-beans/>, 2012-09-29 [cit. 2013-03-08].
- [4] WWW stránky: Platform Versions. <http://developer.android.com/about/dashboards/index.html>, 2013-03-04 [cit. 2013-03-08].
- [5] WWW stránky: App Components. <http://developer.android.com/guide/components/index.html>, [cit. 2013-16-08].
- [6] WWW stránky: App Framework. <http://developer.android.com/about/versions/index.html>, [cit. 2013-16-08].
- [7] WWW stránky: Location and Maps. <http://developer.android.com/guide/topics/location/index.html>, [cit. 2013-16-08].
- [8] WWW stránky: PreferenceActivity. <http://developer.android.com/reference/android/preference/PreferenceActivity.html>, [cit. 2013-18-08].
- [9] WWW stránky: Using Shared Preferences. <http://developer.android.com/guide/topics/data/data-storage.html#pref>, [cit. 2013-18-08].
- [10] WWW stránky: SQLiteDatabase. <http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html>, [cit. 2013-19-08].
- [11] Yadav M.: History of Android. <http://www.tech2crack.com/history-android/>, 2011-30-12 [cit. 2013-03-08].

Příloha A

Obsah CD

- Zdrojové texty programu
- Spustitelný .apk soubor
- Zdrojový tvar písemné práce

Příloha B

Testovací úkoly

1. Vytvořte jednoduchou poznámku s obsahem: test.
2. U vytvořené poznámky libovolně změňte barvu textu a pozadí a průhlednost nastavte na 80%.
3. Změňte styl zadávání místa konání ze zadávání souřadnic na kliknutí na mapu.
4. U vytvořené poznámky si získejte aktuální pozici a zadejte pozici konání(střed Brna).
5. Takto upravenou poznámku vložte na plochu jako widget.
6. Vložte na plochu widget velikosti 2x2.
7. Pomocí tohoto widgetu změňte nastavení tak, aby se v seznamu poznámek nezobrazovaly ukončené poznámky.
8. Vytvořte poznámku s upozorněním na aktuální datum a libovolný čas.
9. Smažte poznámku test.