

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANOTACE SOUBORŮ PDF PRO ANDROID

BAKALÁŘSKÁ PRÁCE

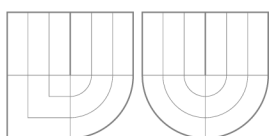
BACHELOR'S THESIS

AUTOR PRÁCE

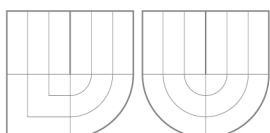
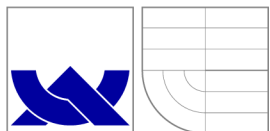
AUTHOR

ONDŘEJ VEJTASA

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ANOTACE SOUBORŮ PDF PRO ANDROID

PDF ANNOTATOR FOR ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ VEJTASA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2012

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací aplikace umožňující anotaci PDF dokumentů na mobilních zařízeních pro platformu Android. Tato aplikace umožňuje nejenom PDF dokumenty prohlížet, ale také do těchto dokumentů vkládat své vlastní poznámky a libovolně s nimi manipulovat. Aplikace je naimplementována v programovacím jazyce Java a využívá moderních knihoven pro práci s PDF dokumenty.

Abstract

The aim of this bachelor thesis is to design and implement an application for Android platform that enables the annotation of PDF documents. This application allows both to read a PDF document and to add and edit customized notes in a PDF document. The application is implemented in Java programming language and makes use of modern libraries for work with PDF documents.

Klíčová slova

PDF anotace, Android, mobilní aplikace, Java, SQLite, APV PDF Viewer, iText, MuPDF.

Keywords

PDF annotation, Android, mobile application, Java, SQLite, APV PDF Viewer, iText, MuPDF.

Citace

Ondřej Vejtasa: Anotace souborů PDF pro Android, bakalářská práce, Brno, FIT VUT v Brně, 2012

Anotace souborů PDF pro Android

Prohlášení

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

.....
Ondřej Vejtasa
13. května 2012

Poděkování

Tímto bych rád poděkoval panu Ing. Zbyňku Křivkovi, Ph.D., vedoucímu bakalářské práce, za odbornou pomoc, ochotu a čas, který mi při tvorbě této práce věnoval.

© Ondřej Vejtasa, 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	3
2 Teorie	4
2.1 Android	4
2.1.1 Historie systému Android	4
2.1.2 Architektura systému Android	6
2.2 Vývoj aplikací pro Android	8
2.3 Komponenty aplikace	8
2.3.1 Aktivity (Activities)	8
2.3.2 Služby (Services)	10
2.3.3 Poskytovatelé obsahu (Content Providers)	10
2.3.4 Příjímače broadcastu (Broadcast receivers)	11
2.3.5 Aktivace komponenty	11
2.3.6 Uživatelské rozhraní	11
2.3.7 Android Manifest	12
2.4 Google Play (dříve Android Market)	12
2.5 PDF	13
2.6 Knihovna iText	14
2.7 APV PDF Viewer	15
3 Návrh aplikace	16
3.1 Cíle aplikace	16
3.1.1 Přehled už existujících řešení	16
3.2 Základní požadavky	17
3.3 Využití existující aplikace	18
3.4 Uživatelské rozhraní	18
3.5 PDF dokument	19
3.5.1 Objekt poznámky v PDF dokumentu	19
3.5.2 Převod souřadnic	21
3.5.3 Zobrazení dokumentu v počítači	21
3.6 Konceptuální model databáze	22
4 Implementace	23
4.1 Balíčky	23
4.2 Databáze	23
4.3 Poznámky	24
4.3.1 Vykreslení poznámek	24
4.4 Aktivity	24

4.5	Texty aplikace	26
4.6	Uložení PDF dokumentu	27
5	Testování	29
5.1	Android emulátor	29
5.2	Fyzická zařízení	30
5.3	PDF dokument	31
5.4	Publikování aplikace	31
6	Závěr	32
6.1	Přínos práce	32
6.2	Budoucnost projektu	32
A	Obsah CD	35
B	Uživatelská příručka	36
B.1	Výběr dokumentu	36
B.2	Vkládání poznámek	37
	B.2.1 Textová poznámka	37
	B.2.2 Oválná a čtvercová poznámka	38
B.3	Čtení a úprava poznámek	38
	B.3.1 Čtení obsahu poznámky	38
	B.3.2 Úprava poznámky	38
B.4	Seznam poznámek	39
B.5	Uložení dokumentu	39
B.6	Zobrazení dokumentu v počítači	40

Kapitola 1

Úvod

Není tomu tak dávno, kdy se na trhu s mobilními telefony objevilo první zařízení disponující dotykovou obrazovkou. Dotykové ovládání přineslo nové možnosti využití mobilního telefonu v běžném životě. Postupně se začaly vyvíjet i nové operační systémy, které jsou přímo určeny pro mobilní telefony s dotykovým ovládáním. Tyto systémy se vyznačují zejména jednoduchým a intuitivním ovládáním, přehledností a rychlostí, ale také volností pro vývojáře mobilních aplikací, kterým systémy umožňují plně využít výpočetní výkon těchto zařízení.

S nástupem operačních systémů, jako jsou např. Google Android, Apple iOS či Windows Phone bylo třeba zajistit centralizovanou distribuci aplikací pro koncové uživatele. K těmto účelům vznikly služby Google Play (dříve Android Market), App Store nebo Marketplace, které umožňují uživateli pohodlně získat téměř jakoukoli aplikaci z velmi rozsáhlé nabídky aplikací a her určených přímo pro zařízení konkrétního uživatele. Služba Google Play umožňuje mimo jiné i jednoduchý nákup elektronických knih a materiálů ve formě PDF dokumentů, které se stávají mezi uživateli stále oblíbenějšími.

Cílem bakalářské práce je navrhnout a vytvořit aplikaci, která umožňuje zobrazení obsahu PDF dokumentu a pohodlnou práci s poznámkami. Problematika zobrazení, vkládání či úprav poznámek PDF dokumentů je velmi zajímavá a využitelnost aplikace v praxi poměrně vysoká, neboť v dnešní době používá velké procento uživatelů dotykových obrazovek svých mobilních zařízení ke čtení elektronických materiálů v podobě PDF dokumentů a v době psaní tohoto textu neexistovala žádná volně dostupná aplikace, která by tuto nebo podobnou funkčnost nabízela.

Kapitola 2 této práce se zabývá teoretickou částí, která popisuje platformu Android a některé její klíčové prvky. V kapitole se také nachází krátké seznámení s multiplatformním formátem dokumentů PDF, knihovnou umožňující práci s PDF dokumenty pod platformou Android a aplikací pro čtení a zobrazení PDF dokumentů.

V kapitole 3 jsou uvedeny cíle a požadavky kladené na aplikaci. Rozebrán je také dopodrobna formát dokumentů PDF a objekt poznámky. Nechybí ani konceptuální návrh modelu databáze, kterou aplikace využívá.

Kapitola 4 popisuje aplikaci z implementačního hlediska. Rozebírá použité postupy, metody a balíčky, které byly při vývoji aplikace použity.

Na závěr se tato práce zabývá testováním vytvořené aplikace v emulátoru a na různých fyzických zařízeních, publikováním aplikace na digitální úložiště aplikací Google Play a dalším možným budoucím vývojem aplikace. Přílohou tohoto textu je také návod na ovládání vytvořené aplikace a obsah příloženého CD.

Kapitola 2

Teorie

Tato kapitola se zabývá popisem platformy Android a její historií. Stručně také popisuje klíčové prvky, které jsou s touto platformou spjaty. V další části kapitoly se nachází seznámení s multiplatformním formátem dokumentů PDF a knihovnou iText pro práci s PDF dokumenty pod platformou Android. Poslední část kapitoly je věnována aplikaci APV PDF Viewer umožňující pohodlné zobrazování PDF dokumentů pod výše zmíněnou platformou.

2.1 Android

Android je platforma s otevřeným zdrojovým kódem, která je určena především pro mobilní zařízení. Obsahuje zejména operační systém, který je založen na jádře operačního systému Linux, uživatelské rozhraní a základní aplikace.

2.1.1 Historie systému Android

Počátky vzniku systému Android sahají do roku 2003, kdy čtveřice vývojářů Andy Rubin, Rich Miner, Nick Sears a Chris White v Kalifornii zakládají společnost Android Inc., která se specializuje na vývoj software pro malá mobilní zařízení. V roce 2005 tuto společnost v tichosti kupuje Google Inc. a stává se jeho dceřinou společností pod vedením Andyho Rubina. V roce 2007 získává Google několik patentů v oblasti mobilních zařízení a potvrzuje tak domněnky odborné veřejnosti o tom, že chce Google vstoupit na trh tzv. chytrých mobilních telefonů.

Dalším významným krokem bylo založení konsorcia Open Handset Alliance (OHA), které vzniklo v roce 2007 a které si kladlo za cíl vyvinout otevřený standard pro mobilní zařízení. Jde o skupinu výrobců mobilních telefonů, telekomunikačních operátorů a technologických firem, které stojí za vývojem operačního systému Android a v jehož čele je společnost Google.

Verze systému Android

Pro laika je asi největší zajímavostí pojmenovávání jednotlivých verzí systému Android, které se řídí podle zákusků (dessert) v abecedním pořadí. Úplně prvními verzemi byly Astro a Bender, nicméně kvůli ochranným známkám se tyto názvy oficiálně nikde neuváděly. Dalšími verzemi jsou: Cupcake (ovocný košíček), Donut (kobliha), Eclair (plněný banánek), Froyo (mražený jogurt), Gingerbread (perník), Honeycomb (medový bonbon) a Ice Cream Sandwich (zmrzlina v čokoládové oplatce).

Během psaní tohoto textu se na trhu objevila nová verze systému Android. Jde o Android 4.0 Ice Cream Sandwich, velmi významnou platformovou verzi. Tato verze přináší mnoho aplikací z Androidu 3.0 Honeycomb, určeného především pro tablety a zařízení s větším displejem do menších zařízení, např. do mobilních telefonů. Vývojář už tedy není nucen vyvíjet dvě odlišné verze své aplikace zvláště pro platformu Android 2.0 a Android 3.0.

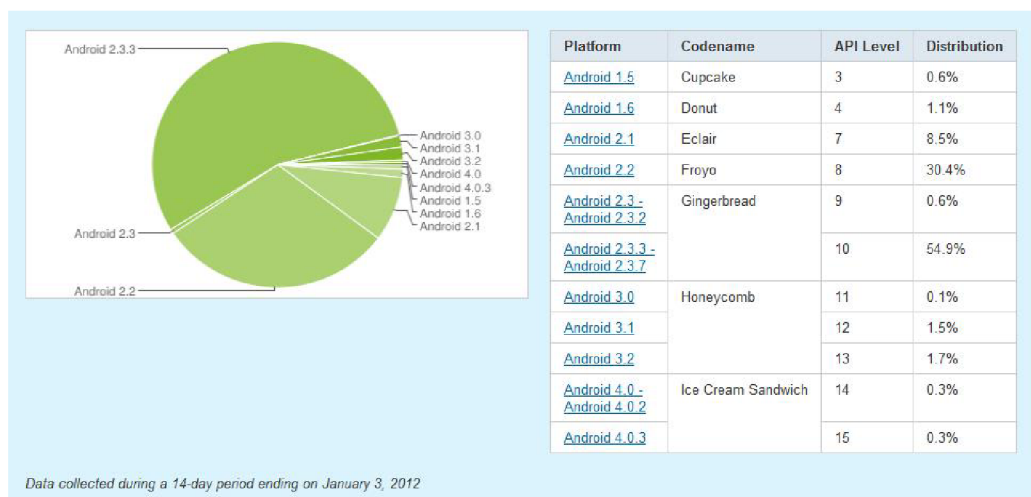
Jednotlivé verze systému Android [24]:

- *1.0* (28. září 2008): Šlo o první oficiální komerční verzi systému Android, která byla představena na zařízení HTC dream (G1). Nechyběla integrace služeb Google, podpora Wi-Fi či Bluetooth či integrovaný webový prohlížeč.
- *1.1* (9. únor 2009): Velkých změn se v této verzi dočkalo zejména uživatelské rozhraní a jeho ovládání. Tato verze byla dostupná pouze pro zařízení T-Mobile G1.
- *1.5 Cupcake* (30. duben 2009): Byla přidána možnost nahrávat a sledovat videa z kamery, stejně tak byla přidána podpora přehrávání 3GP a MPEG-4 videa. Další novinkou byly animované přechody mezi obrazovkami či mnohem rychlejší získání přesné GPS polohy.
- *1.6 Donut* (15. září 2009): Tato verze přinesla nové prostředí pro fotoaparát, kameru a zobrazování obrázků včetně podpory rozlišení obrazovky WVGA. Dalšího vylepšení se dočkal Android Market. Poprvé se zde objevuje indikátor stavu baterie nebo možnost rychlého textového i hlasového vyhledávání.
- *2.0/2.1 Eclair* (26. říjen 2009): Nově umožňovala podporu Bluetooth 2.1. Objevila se i možnost vyhledávat všechny uložené SMS a MMS zprávy a svého vylepšení se dočkalo i uživatelské rozhraní, webový prohlížeč s podporou HTML5 nebo kalendář s novými funkcemi.
- *2.2 Froyo* (20. květen 2010): Nejdůležitější novinkou této verze byla nově podpora Adobe Flash 10.1 spolu se zvýšením rychlosti a výkonu. Nově se zde objevuje možnost instalace aplikací do rozšířené paměti nebo možnost vytvoření přenosného Hotspotu.
- *2.3 Gingerbread* (prosinec 2010): Tato verze přináší nativní¹ podporu pro SIP internetovou VoIP telefonii a vylepšený textový a hlasový vstupní režim. Vývojáři také mysleli na podporu pro více kamer v zařízení, včetně přední VGA kamery. Vylepšení se dočkala také klávesnice a velice používané operace kopírování/vkládání textu.
- *3.x Honeycomb* (22. únor 2011): Jednalo se o první verzi Androidu pouze pro tablet. Prvním přístrojem, na kterém tato verze běžela, byl Motorola Xoom tablet. Tato verze nabízela přepracovanou klávesnici optimalizovanou pro rychlé psaní a nabídla svým uživatelům kompletně přepracované a vylepšené uživatelské rozhraní. Nově se zde objevuje možnost vytvoření Bluetooth tetheringu.
- *4.0 Ice cream sandwich* (19. říjen 2011): Prvním přístrojem podporující tuto platformu byl Samsung Galaxy Nexus. Tato verze platformy by podle společnosti Google měla být „teoreticky“ kompatibilní s libovolným zařízením podporujícím platformu ve verzi 2.3 Gingerbread. S rozmáhajícími se dotykovými displeji byla velmi uvítána zejména podpora virtuálních tlačítek zařízení přímo v uživatelském rozhraní. Zvýšená rychlost a výkon, hardwarová akcelerace uživatelského rozhraní a vestavěný editor fotografií je

¹nativní: vlastní pro danou platformu, specifický pro daný systém.

spolu s automatickou synchronizací záložek internetového prohlížeče Google Chrome další novinkou.

Podíl jednotlivých verzí systému Android na trhu, který ilustruje obrázek 2.1, se velmi rychle mění. Je ovšem velmi dobrým zvykem při vývoji aplikací zachovávat zpětnou kompatibilitu i pro starší zařízení se staršími verzemi platformy Android.



Obrázek 2.1: Podíl jednotlivých verzí Androidu na trhu [8]

2.1.2 Architektura systému Android

Android je postaven na pevném a osvědčeném základu jádra operačního systému Linux, který poskytuje abstraktní hardwarovou vrstvu umožňující Androidu proniknout do široké škály platform budoucnosti. Mezi hlavní komponenty operačního systému Android, jehož architektura je znázorněna na obrázku 2.2, patří:

Aplikace (Applications)

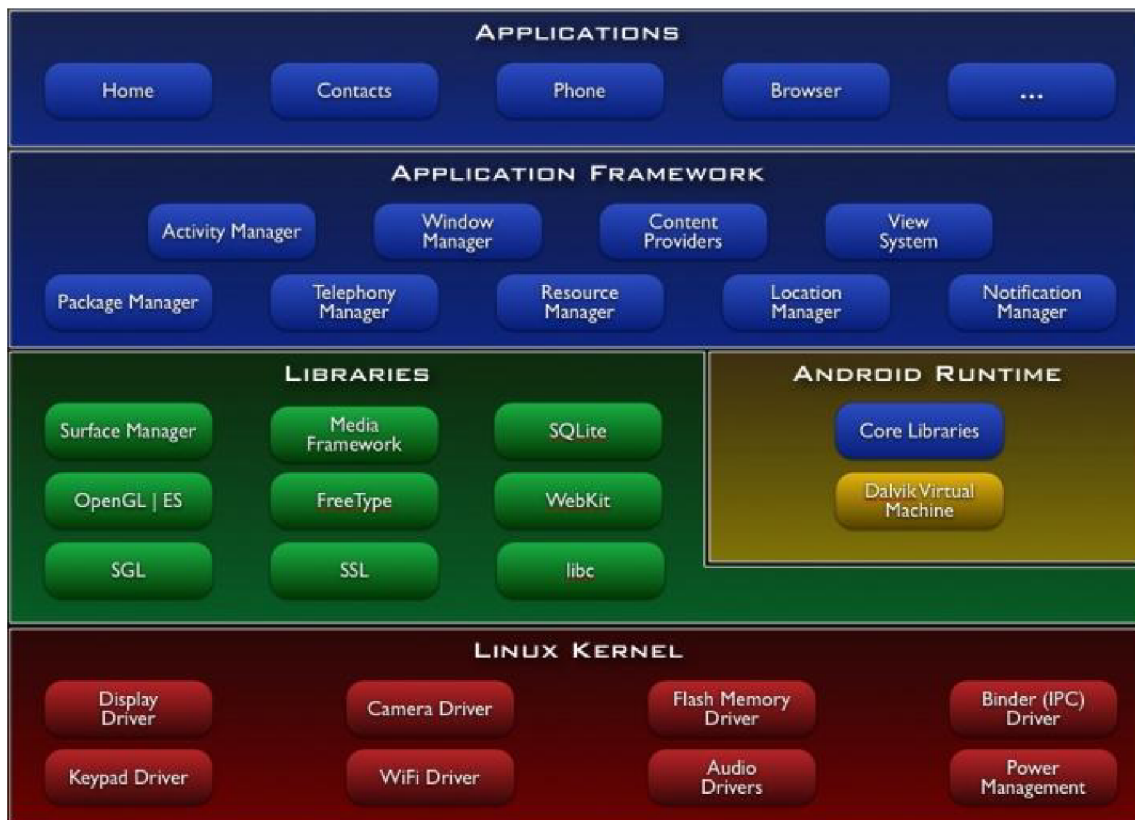
Jde o vrstvu, kterou využívají běžní uživatelé, ve které se nachází aplikace předinstalované nebo nainstalované dodatečně uživatelem. Obsahuje sadu základních aplikací včetně e-mailového klienta, SMS programu, kalendáře, map, prohlížeče, kontaktů a dalších. Všechny aplikace jsou napsány pomocí programovacího jazyka Java.

Aplikační rozhraní (Application Framework)

Poskytuje přístup k velkému množství služeb a částí systému, které lze využít přímo v aplikacích. Architektura aplikací je navržena tak, aby zjednodušila opakované použití komponent. Jakákoliv aplikace může poskytnout své služby dalším aplikacím, které je poté mohou využít (v závislosti na bezpečnostním omezení aplikačního rozhraní).

Knihovny (Libraries)

Android obsahuje sadu C/C++ knihoven, které používají různé komponenty systému. Vývojářům jsou tyto knihovny dostupné prostřednictvím aplikačního rozhraní. Mezi základní knihovny patří:



Obrázek 2.2: Architektura operačního systému Android [12]

- *Systémová C knihovna* – BSD implementace standardní knihovny C, optimalizovaná pro vestavěná zařízení se systémem Linux,
- *Knihovna médií* – podpora přehrávání a nahrávání v mnoha populárních audio a video formátech, zobrazování obrazových souborů,
- *Správa řízení zobrazení* – zobrazení 2D a 3D grafických vrstev,
- *LibWebCore* – knihovna pro webový prohlížeč,
- *SGL* – základní 2D grafická knihovna,
- *3D knihovny* – knihovna pro hardwarovou 3D akceleraci nebo optimalizované 3D rastrování,
- *FreeType* – bitmapové a vektorové vykreslování písma,
- *SQLite* – relační databázový stroj, který je k dispozici pro všechny aplikace.

Prostředí pro běh aplikace (Android Runtime)

Každá spuštěná aplikace běží jako samostatný proces s vlastní instancí virtuálního stroje Dalvik. Dalvik je aplikační virtuální stroj vyvinut speciálně pro Android, který je optimalizován pro efektivní běh více virtuálních strojů a zároveň minimalizuje nároky na paměť a energii. Využívá základní funkce jádra systému, jako použití vláken nebo nízko-úrovňovou správu paměti.

Jádro Linuxu (Linux Kernel)

Samotné jádro vychází z jádra operačního systému Linux ve verzi 2.6, které tvoří abstraktní vrstvu mezi hardwarem a ostatním softwarem ve vyšších vrstvách. Poskytuje základní služby jako je např. zabezpečení, správa paměti, řízení procesů apod. Důvodem, proč bylo zvoleno právě jádro Linuxu, byla možnost snadného sestavení na různých zařízeních a tím zaručena velká míra přenositelnosti.

2.2 Vývoj aplikací pro Android

V případě vyvíjení aplikací, které potřebují co nejrychlejší odezvu (jedná se především o hry a graficky náročnější aplikace s odezvou v reálném čase), se využívá NDK (Native Development Kit). NDK je balík pro vývoj v C/C++ nativním kódem. Nevýhodou programování v nativním kódě je absence některých standardních knihoven a kvalitního prostředí pro programování.

Ostatní aplikace, které jsou méně náročné na výkon, jsou primárně programovány v programovacím jazyce Java za použití SDK². Ten obsahuje jednotlivé nástroje pro překlad, testování a ladění aplikací. Nejedná se ovšem o klasickou verzi jazyka Java, která by vycházela ze standardu Java SE nebo Java ME pro mobilní telefony, a která s těmito standardy není plně kompatibilní. Každá verze systému Android má své specifické programátorské rozhraní (API), které poskytuje programátorovi knihovny, jež může použít. SDK obsahuje také poměrně rozsáhlou a podrobnou dokumentaci popisující jednotlivé knihovny, třídy, metody a popis s příklady použití. Vývoj aplikací pomocí SDK je možný jak pod operačními systémy Windows, tak o pod systémy Linux nebo Mac [13].

2.3 Komponenty aplikace

Každá aplikace je zabalena do jednoho archivu s příponou `.apk`, pomocí něhož je možné aplikaci nainstalovat a distribuovat. Archiv obsahuje mimo přeložených zdrojových souborů aplikace také grafické prvky nebo soubor `AndroidManifest.xml`.

Aplikace pro systém Android se skládá z jedné nebo více aplikačních komponent, kterými jsou *aktivity (activities)*, *služby (services)*, *poskytovatelé obsahu (content providers)* a *příjemce broadcastu (broadcast receivers)*. Každá komponenta plní jinou roli v chování aplikace a každá může být aktivována zcela samostatně i z jiných aplikací [5].

2.3.1 Aktivity (Activities)

Aktivita je součástí aplikace, která nabízí obrazovku s uživatelským rozhraním, se kterou mohou uživatelé komunikovat. Příkladem může být vytočení telefonního čísla, odeslání e-mailové zprávy nebo hledání na mapě. Ve většině případů se jedná o okno zaplňující celou obrazovku, které je umístěno na popředí a se kterým uživatel manipuluje. Pro implementaci je třeba vytvořit podtřídu hlavní třídy `Activity`.

Aplikace se obvykle skládá z několika aktivit, které jsou navzájem provázané. Jedna z aktivit je typicky označována jako *hlavní* a je aktivována při spuštění aplikace. Každá aktivita pak může spouštět další aktivity, které provádějí různé operace.

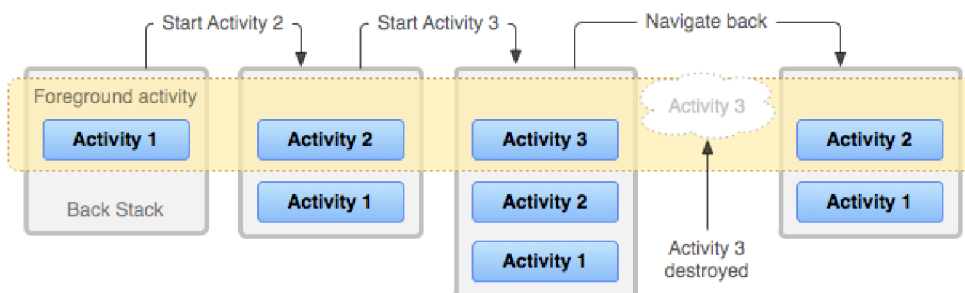
Grafické uživatelské rozhraní aktivity se skládá z hierarchie objektů, které vycházejí ze základní třídy `Views`. Každý objekt ovládá svůj přiřazený obdélníkový prostor v dané

²Software Development Kit: soubor nástrojů pro vývoj aplikace dané platformy.

aktivitě a může reagovat na události vyvolané uživatelem. Příkladem tohoto objektu může být tlačítko, které spustí akci, pokud jej uživatel aktivuje [3].

Životní cyklus aktivity

Aktivity v systému Android jsou řízeny pomocí zásobníku. Při spuštění je nová aktivita umístěna na vrchol zásobníku a stává se aktuální. Předchozí aktivita zůstává na zásobníku pod aktuální aktivitou a čeká, dokud se aktuální aktivita neukončí. Pokud stiskne uživatel tlačítko *Zpět*, aktuální aktivita je ukončena a předchozí aktivita je obnovena [10]. Činnost zásobníku při práci s aktivitami ilustruje obrázek 2.3. Celý životní cyklus je znázorněn na obrázku 2.4.



Obrázek 2.3: Reprezentace činnosti zásobníku při práci s aktivitami [10].

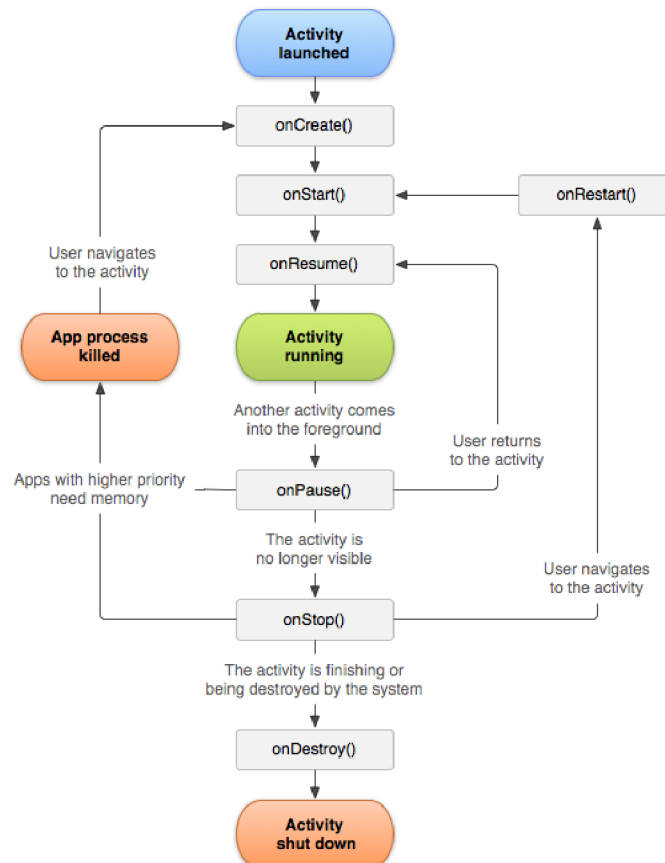
Aktivita se může nacházet ve třech základních stavech:

- *Obnovena (resumed)* – Aktivita je v popředí obrazovky a jsou na ni směřovány akce uživatele. Tento stav se také někdy označuje jako běžící (ang. *running*).
- *Pozastavena (paused)* – Aktivita je pozastavena, pokud dojde k jejímu částečnému překrytí jinou aktivitou, případně je zcela překryta průhlednou aktivitou, která se nevztahuje na celou obrazovku. Pozastavená aktivita je stále uložena v paměti a přístupná přes správce oken. Aktivita může být systémem automaticky ukončena, pokud dojde k extrémně nízkému stavu paměti.
- *Zastavena (stopped)* – K zastavení aktivity dojde tehdy, pokud je zcela překryta jinou aktivitou a není pro uživatele viditelná. Zastavená aktivita je sice stále uchovávána v paměti, ale už není přístupná přes správce oken. Pokud systém potřebuje uvolnit operační paměť, zastavenou aplikaci ukončí.

Pokud dojde k přechodu aktivity do jiného stavu popsaného výše, aktivita o své změně stavu informuje prostřednictvím speciálních metod zpětného volání [3]. Mezi metody informující o změně stavu aktivity patří:

- *onCreate(Bundle savedInstanceState)* – volána při prvním vytvoření aktivity. Parametr *savedInstanceState* slouží k obnově předchozího stavu aktivity,
- *onStart()* – volána předtím, než se aktivita zviditelní uživateli,
- *onResume()* – volána předtím, než dojde k interakci aktivity s uživatelem,

- *onPause()* – volána předtím, než je aktivita pozastavena,
- *onStop()* – aktivita není dále uživatelem viditelná a poté je zastavena,
- *onRestart()* – volána po tom, co byla aktivita zastavena a bude znovu spuštěna,
- *onDestroy()* – volána předtím, než je aktivita zcela ukončena.



Obrázek 2.4: Životní cyklus aktivity [3].

2.3.2 Služby (Services)

Jedná se o službu, která je spuštěna na pozadí bez přímé interakce s uživatelem a neposkytuje grafické uživatelské rozhraní. Pro implementaci této služby je třeba využít podtřídy **Service**. Příkladem této služby může být e-mailový klient automaticky stahující poštu na pozadí nebo přehrávání hudby i v případě, že uživatel pracuje s jinou aplikací, či automatická kontrola aktualizací.

2.3.3 Poskytovatelé obsahu (Content Providers)

Poskytovatel obsahu řídí přístup k uspořádanému souboru dat a umožňuje tak jeho sdílení mezi aplikacemi. Prostřednictvím této komponenty jednotlivé aplikace povolují přístup

jiným aplikacím k manipulaci se svými daty. Tato data mohou být uložena v systému souborů nebo v databázi SQLite, kterou systém Android disponuje. Přistoupit k takto sdíleným datům lze pomocí instance třídy `ContentResolver`. V každé aplikaci je možné upravit přístupová práva ke sdíleným datům v souboru `AndroidManifest.xml`, který je popsán v sekci [2.3.7](#)

2.3.4 Přijímače broadcastu (Broadcast receivers)

Komponenta naslouchající a reagující na zprávy přicházející z celého systému. Příkladem mohou být informace o zhasnutí obrazovky, vybití baterie nebo změna orientace zařízení. Přestože tato komponenta není pro uživatele viditelná, může vytvářet nejruznější upozornění viditelná ve stavovém řádku. K realizaci je nutné implementovat podtřídu `BroadcastReceiver`.

2.3.5 Aktivace komponenty

Tři ze čtyř typů komponent - aktivity, služby a přijímače broadcastu jsou aktivovány prostřednictvím asynchronních zpráv (*intents*), které umožňují propojení dvou komponent za běhu. Pomocí třídy `Intent` může aplikace vyjádřit svůj požadavek na určitou akci.

Objekt typu `Intent` obsahuje několik složek informací. Mezi nejdůležitější složky patří *akce* definující, kterou akci provést, *data*, se kterými akce pracuje a *název komponenty*, která objekt zpracuje.

2.3.6 Uživatelské rozhraní

V aplikacích pro Android se uživatelské rozhraní (UI³) skládá z objektů tříd `View` a `ViewGroup`. Objekty třídy `View` jsou základními jednotkami uživatelského rozhraní.

Třída `View` slouží jako základ pro podtřídy zvané `Widgets`, které nabízejí plně implementované objekty uživatelského rozhraní (např. textová pole či tlačítka). Třída `ViewGroup` je základem pro podtřídy zvané `Layouts`, které řídí rozložení prvků uživatelského rozhraní [11]. Základními typy rozložení jsou:

- *FrameLayout* – jedná se o nejjednodušší typ rozložení objektu. Jde v podstatě o prázdné místo na obrazovce, které je možné později vyplnit jediným objektem, např. obrázkem,
- *LinearLayout* – zarovnává všechny prvky v jednom směru – vodorovně nebo svisle podle toho, jak je definována orientace atributu,
- *TableLayout* – umístí prvky do řádků a sloupců bez zobrazení čar ohraničení,
- *RelativeLayout* – určuje polohu prvku vzhledem k rodičovskému prvku nebo jinému objektu podle zadaného identifikátoru.

Pro definici vzhledu každé komponenty uživatelského rozhraní lze využít jednu ze dvou nabízených možností. První možnost nabízí definovat rozložení prvků jednotlivých komponent pomocí XML souborů, druhá možnost umožňuje vytvořit rozložení prvků přímo za běhu aplikace programově.

³ *User interface: anglická zkratka pro uživatelské rozhraní.*

2.3.7 Android Manifest

Každá aplikace musí obsahovat soubor `AndroidManifest.xml` ve svém kořenovém adresáři. Jedná se o strukturovaný XML⁴ soubor, který obsahuje základní informace o aplikaci, bez kterých systém Android nemůže danou aplikaci identifikovat a spustit [17].

Mezi tyto informace patří:

- *název balíčku* – název balíčku slouží jako jedinečný identifikátor pro aplikaci,
- *komponenty aplikace* – mezi které mimo jiné patří aktivity, služby, poskytovatelé obsahu nebo přijímače broadcastu, kterými se zabývá sekce,
- *práva aplikace* – veškerá systémová oprávnění pro přístup k chráněným částem API⁵, např. přístup k internetu, dále popisuje interakci s komponentami aplikace,
- *název a ikonka aplikace* – včetně deklarace minimální verze API, kterou aplikace vyžaduje ke svému správnému chodu.

Zdrojový kód 2.1 je ukázkou souboru `AndroidManifest.xml` k aplikaci, která požaduje povolení přístupu k ovládání vibrací zařízení a obsahuje vstupní aktivitu `AnnotationActivity`.

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cx.pdf.android.pdfview">
    <uses-permission android:name="android.permission.VIBRATE"/>
    <application android:icon="@drawable/icon" android:label="@string/
        app_name">
        <activity android:name=".AnnotationActivity">
            <Intent-filter>
                <action android:name="android.Intent.action.MAIN"/>
                <category android:name="android.Intent.category
                    .LAUNCHER"/>
            </Intent-filter>
        </activity>
    </application>
</manifest>
```

Zdrojový kód 2.1: Ukázka souboru `AndroidManifest.xml`.

2.4 Google Play (dříve Android Market)

Jedním z nejúčinnějších způsobů, jak dostat aplikaci do rukou uživatelů, je publikování na Google Play. Google Play je digitální úložiště pro aplikace, knihy, filmy a hudbu, které je přístupné všem uživatelům s Android zařízením a komukoli na webu. Začátkem března 2012 Google Play nahradil a rozšířil dřívější Android Market o knihy, hudbu a filmy a nabídl tak uživatelům všechen tento digitální obsah na jednom místě [6].

⁴*Extensible Markup Language*: obecný značkovací jazyk, který je standardizován konsorciem W3C.

⁵*Application Programming Interface*: jedná se o rozhraní pro programování aplikací.

Publikování aplikací

Před publikováním své aplikace je nutná registrace vývojáře na Google Play. Během tohoto procesu registrace je nutné vytvořit profil vývojáře, zaplatit registrační poplatek 25 US\$ a potvrdit distribuční dohodu. Pokud chce vývojář svou aplikaci prodávat, musí mít založen obchodní účet Google Checkout. Google si pak strhává 30% z ceny každé prodané aplikace [9]. Čeští vývojáři se spuštění této služby dočkali až v dubnu 2012, kdy se ČR stala 30. zemí, kde Google tuto možnost nabídl [7].

Bezpečnost

Android Market byl nechvalně proslulý svou nedostatečnou bezpečností a nebylo žádnou výjimkou, že jinak šikovná aplikace s řadou užitečných funkcí obsahovala malware⁶. Tomuto se společnost Google rozhodla zabránit a začátkem měsíce února 2012 spustila novou službu Bouncer. Tato nová bezpečnostní vrstva poskytuje automatizované testování aplikací na potenciálně škodlivý software a nově je také třeba projít procesem schvalování žádosti o vložení nové aplikace do Google Play [15].

2.5 PDF

Jde o přenosný formát dokumentů PDF (Portable Document Format), který slouží pro bezpečnou a spolehlivou výměnu informací nezávisle na platformě.

Historie

Zpočátku šlo pouze o interní projekt společnosti Adobe Systems pod vedením jednoho ze zakladatelů společnosti Johna Warnocka. Cílem tohoto projektu bylo vytvoření takového formátu souboru, aby jej bylo možné rozšířit v celé společnosti a zobrazit jej na libovolném počítači pomocí libovolného operačního systému. Základem tohoto formátu se staly dvě technologie. PostScript jako technologie pro popis dokumentů nezávislá na platformě a Adobe Illustrator jako příklad aplikace, která byla podporována na více platformách a zároveň mohla zobrazit jednoduché dokumenty PostScriptu vytvořené pomocí jiných aplikací. Rozšířením těchto dvou technologií vznikl v roce 1993 nový formát souborů PDF (což je ve skutečnosti upravený a optimalizovaný PostScript) a sada aplikací pro vytváření a zobrazování souborů tohoto tehdy nového formátu [21]. Stručnou historii nejpoužívanějších verzí PDF dokumentů a softwarového nástroje pro jejich úpravu Adobe Acrobat ilustruje tabulka 2.2.

Rok vydání	2000	2001	2003	2005
PDF verze	PDF 1.3	PDF 1.4	PDF 1.5	PDF 1.6
Software	Acrobat 4.0	Acrobat 5.0	Acrobat 6.0	Acrobat 7.0
Rok vydání	2006	2008	2009	2011
PDF verze	PDF 1.7	PDF 1.7 EL 3	PDF 1.7 EL 5	PDF 1.7 EL 8
Software	Acrobat 8.0	Acrobat 9.0	Acrobat 9.1	Acrobat X

Tabulka 2.2: Vývoj formátu PDF dle roku vydání.

⁶Počítačový program určený k proniknutí do systému, který zahrnuje např. počítačové viry, trojské koně, spyware nebo adware.

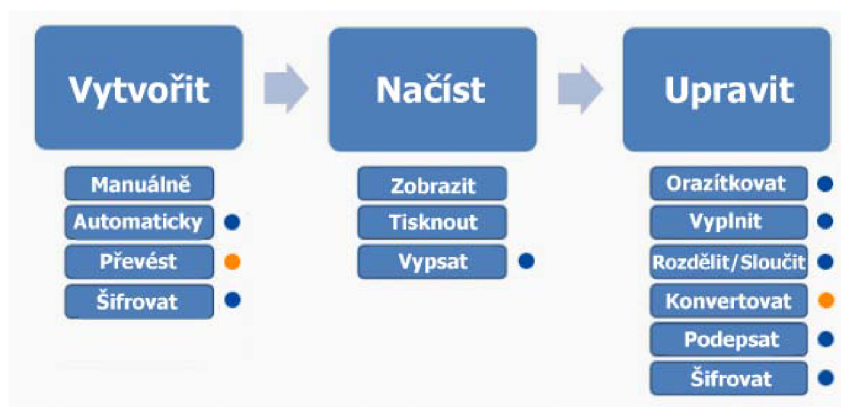
Charakteristika

Od roku 2007 jde o oficiální otevřený standard s označením ISO 32000, který je spravován Mezinárodní organizací pro standardizaci ISO [1]. Nespornou výhodou tohoto formátu dokumentů je zcela jisté podpora nejrůznějších platforem. Díky této podpoře lze PDF dokumenty zobrazit prakticky na jakékoliv platformě, jako např. Microsoft Windows, Mac OS nebo Linux. Podporovány jsou i mobilní platformy Windows Phone, Apple iOS a v neposlední řadě i Google Android. PDF soubory vypadají na každé platformě jako originální dokumenty a uchovávají zdrojové informace ze souboru bez ohledu na aplikace použité k jejich vytvoření nebo zobrazení. Dokumenty ve formátu PDF je možné také digitálně podepisovat či chránit heslem, čímž je zaručena jejich vysoká bezpečnost. PDF formát podporuje také vyhledávání v textu dokumentu, díky čemuž lze dokumenty snadno prohledávat.

2.6 Knihovna iText

iText je knihovna, jejíž hlavním cílem je umožnit vývojářům programově vytvářet a pracovat s PDF dokumenty [18]. Knihovna umožňuje provádět nejrůznější operace s těmito dokumenty, které jsou znázorněny na obrázku 2.5. Operace označené modrým kolečkem jsou plně podporovány knihovnou iText, žlutým kolečkem jsou označeny pouze částečně podporované operace.

Příkladem může být generování PDF dokumentů a zpráv na základě dat načtených z XML souborů nebo databáze, vytváření map a knih využívajících četné interaktivní funkce, které jsou k dispozici ve formátu PDF nebo vyplnění interaktivních formulářů. Mimo jiné knihovna umožňuje rozdělit či spojit stránky z existujících PDF dokumentů, poskytnout dynamicky generované PDF dokumenty, se kterými lze následně manipulovat nebo přidávat čísla stránek, vodoznaky a další vlastnosti k existujícím PDF dokumentům.



Obrázek 2.5: Přehled možných operací s PDF dokumentem.

Přestože je iText knihovna s otevřeným zdrojovým kódem a zdarma, je třeba respektovat licenci Affero General Public License (AGPL)⁷, kterou je chráněna.

⁷Plné znění licence AGPL lze nalézt zde: <http://itextpdf.com/terms-of-use/agpl.php>

2.7 APV PDF Viewer

Jedná se o aplikaci, která umožňuje pohodlné prohlížení obsahu PDF dokumentů pod platformou Android verze 1.5 Cupcake a vyšší. APV je publikována pod licencí GNU GPL verze 3⁸ a je založena na několika následujících knihovnách, pomocí kterých vykresluje obsah PDF dokumentu [14]:

- *Knihovna FreeType2* – Efektivní softwarový nástroj pro generování vysoce kvalitního písma ve formě obrázků [23]. Používá se v nejrůznějších knihovnách pro vytváření grafiky, zobrazování informací na displeji, konverzní nástroje pro práci s písmem nebo generování obrazového textu. Knihovna neposkytuje API rozhraní, jako je např. uspořádání textu a grafické zpracování. Nicméně tyto úkoly výrazně zjednodušuje tím, že poskytuje jednoduché, snadno ovladatelné a jednotné rozhraní pro přístup k obsahu souborů s písmem. Knihovna je vydána pod BSD a GNU GPL v3 licencí.
- *Knihovna MuPDF* – Vykreslující nástroj, který se nachází v MuPDF, je přizpůsoben pro vysokou kvalitu grafického vyhlazení [19]. Text vytištěné stránky na obrazovce je vyobrazen s přesností na zlomky pixelu pro nejvyšší věrnost reprodukce. MuPDF je malý, rychlý a přesto kompletní. Podporuje PDF 1.7 s transparentností, šifrování, hypertextové odkazy, vyhledávání a další. Dokáže zpracovat také OpenXPS dokumenty. Výhodou MuPDF je fakt, že je napsán modulárně. Tato knihovna je uvolněna pod GNU GPL v3 licencí.
- *Knihovna Jbig2dec* – Provádí kompresi JBIG2 formátu obrazu [22]. JBIG2 je určen pro ztrátové a bezztrátové kódování černobílých obrázků na středně vysokém rozlišení s kompresním poměrem 100:1, proto je pro vykreslování PDF dokumentů ideální. Jedinou nevýhodou je fakt, že je stále ve fázi alfa, což znamená, že se během dekódování mohou objevit určité komplikace.
- *Knihovna OpenJPEG* – Knihovna OpenJPEG [20] je kodek JPEG 2000 napsaný v jazyce C. Byl vyvinut s cílem podpořit využívání JPEG 2000 formátu, nového standardu Photographic Experts Group (JPEG). Kromě základního kodeku jsou ve vývoji knihovny další funkce, jako např. i JP2 a MJ2 (Motion JPEG 2000) formáty souborů. Knihovna je uvolněna pod BSD licencí.
- *Knihovna PDFView* – Bohužel se tato knihovna dále už nevyvíjí, nicméně stále přináší užitečnou podporu při zobrazování PDF dokumentů, jako je např. integrace L^AT_EXu, rozpoznání, zda byl dokument změněn během čtení jinou aplikací, možnost přiblížení a oddálení dokumentu, otáčení stránek apod. Mnoho dalších funkcí této knihovny je ale podporováno pouze produkty společnosti Apple a proto je není možné využít v operačním systému Android.

⁸Plné znění licence GNU GPL verze 3 lze nalézt zde: <http://www.gnu.org/licenses/gpl.html>

Kapitola 3

Návrh aplikace

K návrhu aplikace zcela neodmyslitelně patří prostudování už existujících řešení a stanovení cílů, na základě kterých je třeba určit požadavky na aplikaci. Je také potřeba zvolit vhodný už existující prohlížeč PDF dokumentů, na kterém by byla aplikace založena, verze platformy Android, které bude aplikace podporovat, stanovit vzhled a intuitivitu uživatelského rozhraní a dopodrobna se seznámit s PDF dokumenty a objekty poznámek.

3.1 Cíle aplikace

Cílem aplikace je pohodlné zobrazení obsahu PDF dokumentu a zejména pak vkládání poznámek buď přímo do PDF dokumentu, případně do pomocného souboru. PDF dokument s vloženými poznámkami je třeba pohodlně zobrazit i na stolním počítači. Aplikace musí být schopna nejenom poznámku zobrazit, ale také upravit část textu poznámky, vložit nový text či poznámku úplně odstranit.

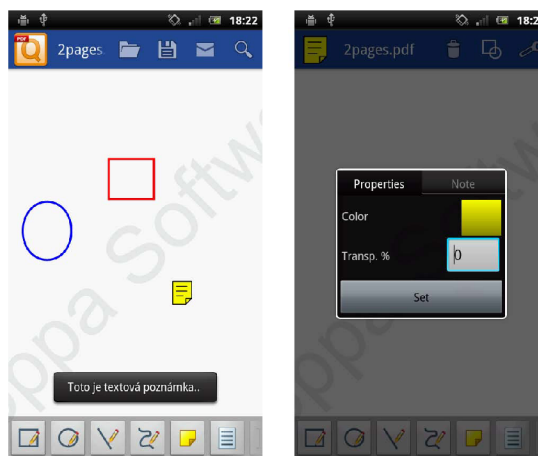
3.1.1 Přehled už existujících řešení

Vzhledem ke skutečnosti, že v době psaní tohoto textu neexistovala žádná volně dostupná aplikace, která by tuto nebo podobnou funkčnost nabízela, bylo nutné prostudovat alespoň demonstrační verzi placené aplikace, která nabízela podobnou funkčnost.

qPDF Notes

Aplikace slouží pro zobrazení a vkládání komentářů do PDF dokumentů a vyplňování interaktivních formulářů. Tato aplikace je nejúspěšnější na trhu. Její hlavní nevýhodou je skutečnost, že je zpoplatněna poměrně vysokým poplatkem¹. Této aplikaci také chybí rozlišování různých druhů a vzhledů ikon textových poznámek. U poznámky nelze vložit autora a předmět. V demo verzi aplikace [16] je možné využívat veškeré dostupné nástroje a PDF dokument libovolně upravovat, ovšem při uložení je přes každou stránku vložen poměrně velký vodoznak, který velmi znepříjemňuje čtení PDF dokumentu. Vložení poznámky do PDF dokumentu v této aplikaci je znázorněno na obrázku 3.1.

¹Cena aplikace qPDF Notes Full byla ke dni 9. května 2012 9.99 US\$.

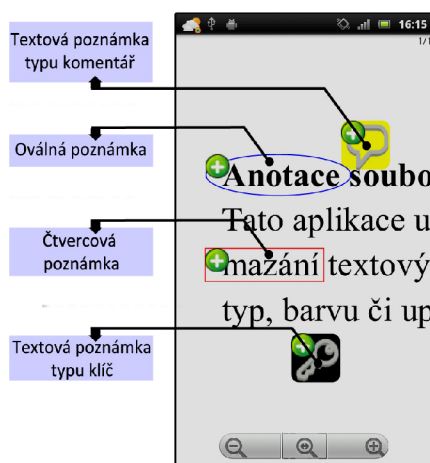


Obrázek 3.1: Ukázka zobrazení poznámek a jejich úprava v aplikaci qPDF Notes.

3.2 Základní požadavky

Po vyzkoušení a prostudování již existující aplikace byly stanovy nové cíle, které musí aplikace splňovat. Aplikace nevyužívá žádného pomocného souboru, jehož využití nabízí zadání této práce, ale veškeré úpravy poznámek zapisuje přímo do PDF dokumentu prostřednictvím knihovny *iText* zmíněné v sekci 2.6.

Mezi podporované typy poznámek patří *textová poznámka*, která je v PDF dokumentu reprezentována ikonkou se specifickým vzhledem a barvou, *oválná poznámka* a *čtvercová poznámka* se zvolenou barvou rámečku.



Obrázek 3.2: Podporované typy poznámek.

U textových poznámek má uživatel možnost zvolit typ a barvu pozadí ikonky, která bude poznámku reprezentovat. Každý prohlížeč PDF dokumentů má předdefinovaný svůj vlastní vzhled ikonek podle jejich typu. Mezi základní typy ikonek, které aplikace podporuje a které jsou v oficiálním standardu PDF dokumentů [2] označovány jako jména (*names*), patří komentář (*Comment*), klíč (*Key*), poznámka (*Note*), nápověda (*Help*), nový odstavec

(*NewParagraph*), odstavec (*Paragraph*) a vsuvka (*Insert*).

Pokud jde o oválnou (*Circle*) nebo čtvercovou (*Square*) poznámku, jedná se o průhledný objekt s barevným rámečkem, jehož barvu si může uživatel zvolit.

Nově vloženou poznámku lze také libovolně přesunout, ovšem pouze v rámci aktuální stránky. U oválných a čtvercových poznámek je možné, aby uživatel změnil libovolně jejich velikost.

Vzhledem k rychlému vývoji platformy Android byla stanovena kompatibilita od verze 2.1 Eclair, čímž je aplikace podle obrázku 2.1 kompatibilní s 98,3% zařízení. Aplikace je primárně určena pro mobilní telefony, které disponují displejem menších rozměrů. Proto je třeba dbát na pohodlnost ovládání aplikace a zobrazovat text přiměřené velikosti. Je však zohledněno používání aplikace také na tabletech.

3.3 Využití existující aplikace

Pro systém Android existuje několik open-source² aplikací, které umožňují zobrazovat obsah PDF dokumentů. Mezi ty nejzdařilejší patří:

- *APV PDF Viewer* – zřejmě nejpropracovanější open-source aplikací založené na knihovně *MuPDF* a pro účely doimplementování podpory práce s poznámkami v PDF dokumentech zcela ideální. Touto aplikací se zabývá sekce 2.7.
- *DroidReader* – vhodný pro začínající vývojáře v NDK (Android Native Development Kit) v projektech Android. Tato aplikace využívá nativních knihoven *libjpeg*, *FreeType*, *MuPDF*, zmíněných v sekci 2.7.
- *VuDroid* – dříve *DjvuDroid*, jedná se o aplikaci založené na kódu knihovny *DjvuDroid* a prohlížeče *PdfDroid*, které se již dále nevyvíjejí.

APV PDF Viewer

APV PDF Viewer byla vybrána jako nejzdařilejší a nejpropracovanější aplikace pro zobrazování PDF dokumentů zejména z důvodu stálého vývoje a podpory aplikace, licenčních podmínek použití a knihovny *MuPDF*, kterou využívá. Tento prohlížeč PDF dokumentů umožňuje veškerou funkcionalitu, kterou uživatel od aplikace tohoto typu očekává. Jedná se například o zobrazení více PDF stránek najednou, snadné vyhledávání v dokumentu, přechod na zvolenou stránku nebo automatické otáčení dokumentu dle polohy zařízení. Další výhodou této aplikace je integrovaný adresářový prohlížeč pro snadný výběr prohlíženého PDF dokumentu. Nevýhodou této aplikace je ovšem fakt, že nepodporuje multi-touch, tedy oddálení nebo přiblížení dokumentu dvěma prsty.

3.4 Uživatelské rozhraní

Vzhledem k tomu, že aplikace vychází z již existujícího prohlížeče PDF dokumentů s vlastním uživatelským rozhraním, bylo nutné uživatelské rozhraní rozšířit o možnost pracovat s PDF poznámkami a ukládání PDF dokumentů. Při návrhu rozšíření uživatelského rozhraní byl kladen důraz zejména na intuitivnost ovládání, přehlednost ovládacích prvků a

²Jde o počítačový software s otevřeným zdrojovým kódem. Při dodržení jistých licenčních podmínek lze kód legálně prohlížet či upravovat.

zachování kompatibility se zařízeními disponujícími s různými velikostmi obrazovky. Nutnou podmínkou byla také přizpůsobitelnost uživatelského rozhraní pro vertikální či horizontální natočení obrazovky. Uživatelské rozhraní disponuje s několika ikonami, které jsou převzaty nebo přímo vycházejí z běžně používaných ikon menu systému Android.

3.5 PDF dokument

Zobrazení PDF dokumentu v aplikaci má na starosti nativní knihovna *MuPDF*. Knihovna sama o sobě neumí zpracovávat a zobrazovat poznámky v PDF dokumentech, které aplikace vytváří. Proto byla tato funkčnost doimplementována s využitím knihovny *iText*.

PDF dokument je rozdělen na jednotlivé stránky, ve kterých je nutné pomocí klíče `/Annots`, který je reprezentován polem objektů, uvést všechny objekty poznámek na dané stránce. V případě smazání poznámky je nutné objekt dané poznámky z tohoto klíče odstranit.

3.5.1 Objekt poznámky v PDF dokumentu

Poznámka je v PDF dokumentu reprezentována jako jednoduchý PDF objekt, jehož syntaxe je znázorněna pomocí následujícího zdrojového kódu 3.1. Objekt obsahuje několik důležitých klíčů, mezi které patří:

- `/Type` – typ objektu, v případě poznámky obsahuje klíč `/Annot`,
- `/Subtype` – podtyp objektu, tedy poznámky. Aplikace podporuje tři základní, kterými jsou *Text*, *Circle* a *Square*,
- `/Name` – název objektu v případě textové poznámky určuje vzhled ikonky, kterou je poznámka reprezentována.

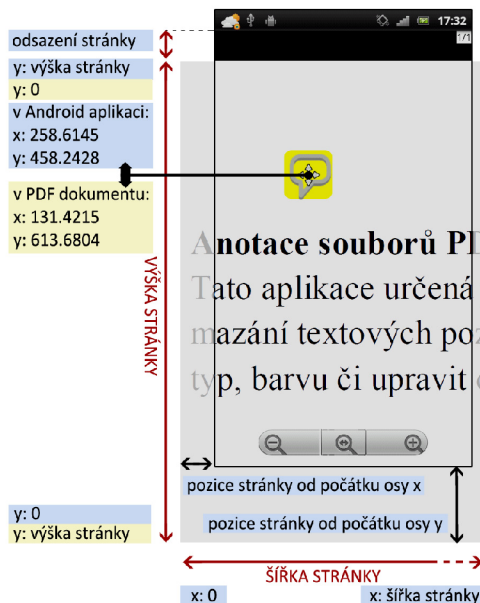
```
1 0 obj
  <</C[1 1 0]/Type/Annot/Subtype/Text/Name/Comment
    /T(autor poznámky)/Subj(Předmět poznámky)/Contents(Samotný obsah poznámky)
    /Rect[334.02 159.58 354.02 179.58]/M(D:20120404112640+02'00)/P 2 0 R>>
endobj
```

Zdrojový kód 3.1: Ukázka objektu poznámky v PDF dokumentu.

Textový obsah poznámky

Samotný textový obsah poznámky mají na starosti následující tři klíče, jejichž obsah není omezen maximálním počtem textových znaků:

- `/T` – autor poznámky,
- `/Subj` – předmět poznámky,
- `/Contents` – obsah poznámky.



Obrázek 3.3: Zobrazení PDF stránky v aplikaci.

Datum poslední změny poznámky

Datum, kdy byla poznámka naposledy upravena, určuje klíč /M. Hodnota klíče musí být ve standardním formátu data dle mezinárodního standardu ASN. 1³. Obsah klíče je textový řetězec ve tvaru (D:YYYYMMDDHHmmSSOHH'mm), kde:

- YYYY – vyjadřuje rok,
- MM – vyjadřuje měsíc (01 – 12),
- DD – vyjadřuje den (01 – 31),
- HH – vyjadřuje hodiny (00 – 23),
- mm – vyjadřuje minuty (00 – 59),
- SS – vyjadřuje sekundy (00 – 59),
- 0 – vyjadřuje časový posun vzhledem k Mezinárodnímu času (UT, Greenwichský čas). Posun je reprezentován jedním z těchto znaků: znaménko plus (+), znaménko mínus (-) nebo velké latinské písmeno Z,
- HH – absolutní hodnota posunu času od UT v hodinách (00 – 23), za touto hodnotou následuje apostrof,
- mm – absolutní hodnota posunu času od UT v minutách (00 – 59).

³Abstract Syntax Notation One: standardní zápis popisující pravidla pro kódování a přenos dat v počítačových sítích.

Pozice poznámky

V PDF dokumentu je pozice objektu, v našem případě poznámky, určena pomocí klíče `/Rect`, který obsahuje pole čtyř prvků. Tyto prvky reprezentují souřadnice jednotlivých bodů pozičního obdélníku v podobě `/Rect[l1x l1y urx ury]`. První dvě souřadnice `l1x` a `l1y` reprezentují levý dolní bod obdélníku, zbylé dvě souřadnice `urx` a `ury` pak pravý horní bod obdélníku na osách `x` a `y`. Tyto body určují pozici poznámky vzhledem k předem určené stránce PDF dokumentu, nikoliv k PDF dokumentu jako celku. Velikost poznámky je určena vzdáleností těchto dvou bodů.

Barva poznámky

Barvu poznámky určuje klíč `/C`. V případě textové poznámky jde o barvu pozadí, u oválných a čtvercových poznámek určuje tento klíč barvu rámečku, kterým jsou ohraničeny. Aplikace podporuje pouze RGB⁴ model, který je v PDF dokumentu reprezentován polem o třech prvcích ve tvaru `/C[R G B]`. Význam jednotlivých prvků je zřejmý, první prvek `R` určuje zastoupení červené složky, druhý prvek `G` zelené složky a třetí prvek `B` modré složky. Zastoupení jednotlivých barev je vyjádřeno hodnotou prvku, která nabývá hodnot v intervalu `0.0 – 1.0`.

Šířka rámečku

U oválných a čtvercových poznámek je třeba určit také šířku rámečku, která se nachází v klíči `/Border`. Klíč je reprezentován polem o několika prvcích ve tvaru `/Border[H V W [D G]]`. První a druhý prvek pole určuje horizontální a vertikální poloměr rohu rámečku, třetí prvek pak hledanou šířku rámečku. Čtvrtým volitelným prvkem je pole, které určuje vzhled rámečku.

3.5.2 Převod souřadnic

Protože systém Android používá odlišné vypočítávání souřadnic konkrétních bodů než je použito v PDF dokumentu, je třeba nejprve tyto souřadnice převést. V PDF dokumentu je počátek souřadných os v levém horním rohu, kdežto v uživatelském rozhraní systému Android je za počátek brán levý dolní roh.

Dalším problémem při výpočtu souřadnic bodu je skutečnost, že PDF stránku nemusíme zobrazovat od jejího levého horního rohu, nýbrž si ji můžeme posunout či přiblížit. V takovém případě je třeba vypočítat skutečnou pozici PDF stránky od počátku osy `x` a `y` a vykreslované body přepočítat. Tento problém je ilustrován na obrázku 3.3.

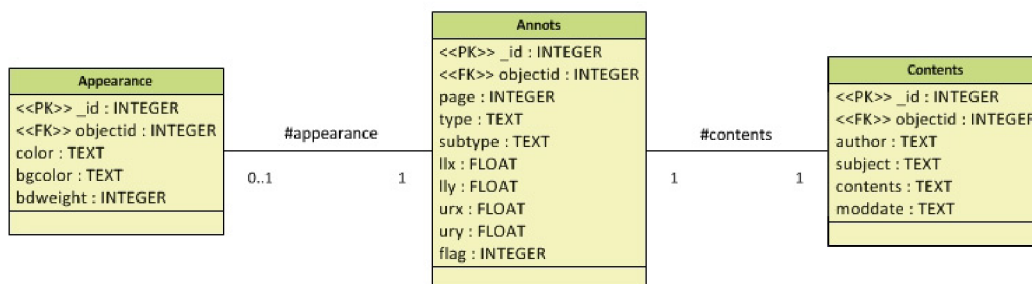
3.5.3 Zobrazení dokumentu v počítači

Správné zobrazení poznámek vložených do PDF dokumentu pomocí aplikace PDF Annotation byl jeden ze stěžejních bodů návrhu. Jedna se zejména o přesný výpočet souřadnic poznámky uvedený v sekci 3.5.2, o korektní registraci objektu poznámky na dané stránce PDF dokumentu, ale také o vyřešení problémů s texty obsahujícími diakritiku.

⁴ RGB barevný model: barevný prostor složený ze tří barevných kanálů Red (červená), Green (zelená) a Blue (modrá), kde je každá barva zastoupena hodnotou své mohutnosti, které může nabývat. Jedná se o aditivní způsob míchání barev.

3.6 Konceptuální model databáze

Systém Android nabízí plnou podporu výkonné souborové relační databáze SQLite, kterou aplikace využívá pro uložení perzistentních dat. Mezi tato data patří veškeré informace o načtených poznámkách z PDF dokumentu. Díky databázi je možné pracovat s poznámkami bez okamžitého zápisu změn do prohlíženého dokumentu. Vztahy jednotlivých entit reprezentujících data uložená v databázi znázorňuje diagram na obrázku 3.4. Diagram obsahuje následující entitní množiny:



Obrázek 3.4: Diagram vztahů mezi entitami.

- *Annots* – Atributy této množiny reprezentují umístění poznámky v PDF dokumentu, její typ a podtyp. Nechybí ani příznak určující aktuální stav poznámky (např. nově vložená, upravená či smazaná).
- *Appearance* – Reprezentuje vzhled každé poznámky. V první verzi této aplikace jde jen o barvu pozadí či popředí poznámky a o barvu rámečku. Do budoucna by mohla být tato množina rozšířena o atributy jako např. šířka rámečku, průhlednost poznámky nebo vržený stín.
- *Contents* – Obsahuje textové údaje poznámky, jako je autor poznámky, předmět, datum poslední změny a samotný text.

Protože může mít každá poznámka maximálně jeden vzhled, je mezi entitními množinami *Appearance* a *Annots* vztah 1..0:1. Každá poznámka má jasně daný svůj konkrétní textový obsah, proto je mezi entitními množinami *Appearance* a *Contents* vztah 1:1.

Primární klíč *_id* slouží jako jedinečný identifikátor pro každý záznam uložený v rámci aplikace. Sekundární klíč *objectid* slouží jako jedinečný identifikátor objektu s poznámkou v rámci načteného PDF dokumentu. V případě vzniku nového záznamu, tedy např. vložení nové poznámky, je tento identifikátor přidělován až při samostatném ukládání PDF dokumentu.

Kapitola 4

Implementace

Tato kapitola se zabývá samotnou implementací aplikace, popisuje použité postupy a metody, které byly při vývoji použity.

Pro implementaci aplikace bylo zvoleno vývojové prostředí Eclipse, které plně podporuje vývoj aplikací pro platformu Android. Implementace probíhala v programovacím jazyce Java s využitím Android SDK, který je zmíněn v sekci 2.2. Při vývoji pomocí těchto prostředků je možné využít Android Emulátoru a vytvořenou aplikaci okamžitě vyzkoušet nebo připojit své vlastní mobilní zařízení s platformou Android a vytvářenou aplikaci testovat přímo na tomto zařízení.

4.1 Balíčky

APV PDF Viewer, ze kterého aplikace vychází, obsahuje dva balíčky. Tyto balíčky byly pro účely aplikace PDF Annotation upraveny a doplněny o implementaci umožňující práci s poznámkami. Balíčky, které mimo jiné slouží jako jedinečný identifikátor aplikace na digitálním úložišti Google Play, bylo nutné přejmenovat. Konkrétně z `cx.hell.android` na `cx.pdf.android`. Důvod je prostý. Bylo nutné od sebe odlišit aplikace APV PDF Viewer a PDF Annotation na tomto úložišti. První balíček, `cx.pdf.android.lib.pagesview`, obsahuje pomocné třídy pro vykreslení jednotlivých dílčích částí PDF dokumentu, výsledků při vyhledávání, ale také třídu pro přístup k SQLite databázi. Druhý balíček, `cx.pdf.android.pdfview`, obsahuje třídy pro práci s aktivitami aplikace a pro práci s PDF dokumentem jako např. propojení s nativními knihovnamí, adresářový manager, správu záložek, ale hlavně třídy pro práci s PDF poznámkami.

4.2 Databáze

Vytvoření databáze dle konceptuálního modelu na obrázku 3.4 je implementováno pomocí třídy `AnnotSQL`, která je založena na speciální rodičovské třídě `SQLiteOpenHelper`. Po otevření zvoleného PDF dokumentu jsou pomocí třídy `Annotation`, která umožňuje získat instanci speciální třídy `SQLiteDatabase` pro vytváření databázových dotazů, extrahovány všechny nalezené poznámky nacházející se v PDF dokumentu. Tyto poznámky jsou posléze vloženy do vytvořené databáze. Veškerá databázová data jsou uložena v souboru `annots.db`, který obsahuje záznamy všech poznámek aktuálně otevřeného PDF dokumentu.

4.3 Poznámky

Poznámky je nejdříve nutné extrahovat z načteného PDF dokumentu. Samotné extrahování je implementováno pomocí třídy `Annotation`, která k tomuto účelu využívá knihovny *iText* zmíněné v kapitole 2.6. Knihovna disponuje třídou `PdfReader`, která umožňuje přistupovat k libovolným objektům a jejich klíčům. Pomocí instance této třídy jsou ve smyčce postupně načítány stránky po stránce klíče `/Annots` s referencemi na PDF objekty s poznámkami. Nalezené objekty jsou poté extrahovány a hodnoty jejich klíčů ukládány. Pokud je objekt s konkrétní poznámkou celý načten, jsou jeho data exportována do databáze SQLite a záznam s poznámkou tak uložen.

4.3.1 Vykreslení poznámek

Poté, co jsou všechny poznámky načteny a uloženy do databáze, probíhá samotné vykreslování extrahovaných poznámek pomocí metody `drawAnnotsPage` třídy `PagesView`. Podle čísla stránky, která je uložena jako hodnota atributu `page` jednotlivých záznamů, jsou načtena všechna data s poznámkami právě zobrazované stránky z SQLite databáze. Následně je nutné přepočítat jejich pozice vzhledem k hodnotě přiblížení a polohy zobrazovaného výřezu stránky.

Textové poznámky jsou reprezentovány oválem, jehož barva výplně odpovídá barvě poznámky. Na ovál je pak umístěna bitmapová ikonka poznámky podle typu. Velikost oválu i ikonky poznámky se dynamicky mění podle hodnoty přiblížení právě prohlížené stránky. U oválných či čtvercových poznámek je vykreslován pouze rámeček. Barvu a šířku rámečku určují atributy `color` a `bdwidth` konkrétního databázového záznamu poznámky.

Rozlišení stavu poznámek

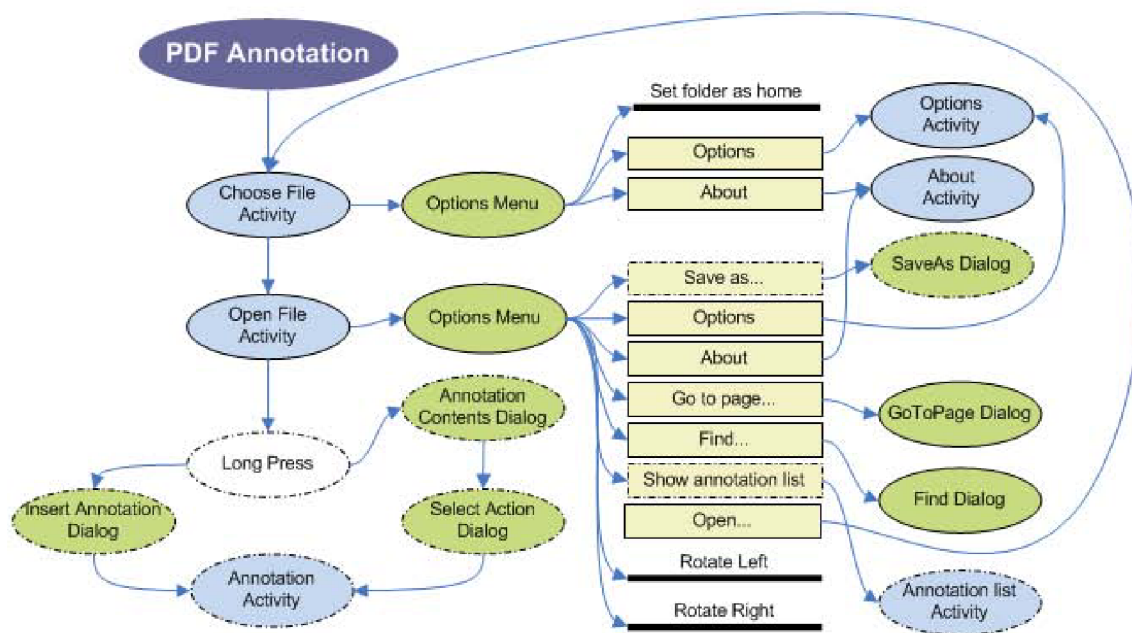
Všechny poznámky jsou označeny malou bitmapovou ikonkou, která reprezentuje stav poznámky. Velikost ikonky je dynamicky vypočítávána na základě hodnoty přiblížení či oddálení zobrazované stránky s poznámkou. Každý databázový záznam obsahuje atribut `flag`, který nabývá hodnot určujících aktuální stav poznámky. Při prvotním načtení všech objektů poznámek z PDF dokumentu mají všechny záznamy hodnotu tohoto atributu 0.

V případě hodnoty 0 je zobrazena ikonka ilustrující modrou vlajku, která symbolizuje uloženou poznámku v PDF dokumentu, která nebyla v aplikaci nijak změněna. Hodnotu 1 znázorňuje červená vlajka symbolizující poznámku načtenou z aktuálního PDF dokumentu, která byla v aplikaci změněna. Záznam může také nabývat hodnoty 2, kterou zastupuje ikonka bílého plus v zeleném poli v případě, že poznámka byla nově vložena v aplikaci a v načteném PDF dokumentu se předtím nenacházela. Poslední možnou hodnotou, kterou může nabývat tento záznam, je hodnota 3 zastupující poznámku určenou k odstranění. Původně byla taková poznámka označena ikonkou bílého mínus v červeném poli, ale v rámci optimalizace aplikace je tato poznámka uživateli zneviditelněna. Fyzicky je poznámka odstraněna až během ukládání změn v PDF dokumentu.

4.4 Aktivity

Vzhled aktivit uživatelského rozhraní je vytvořen deklarativně pomocí samostatných souborů jazyka XML odděleně od aplikační logiky. Obsah aktivit je generován dynamicky přímo ze zdrojového kódu na základě záznamů načtených z databáze SQLite.

Diagram na obrázku 4.1 zobrazuje jednotlivé aktivity, dialogy a vztahy mezi nimi. Modrou barvou jsou znázorněny aktivity, zelenou pak dialogy a hlavní menu aplikace. Událost vyvolávající nové dialogy je reprezentována bílou barvou. Jednotlivé položky hlavního menu ilustruje barva žlutá. Prvky diagramu, které jsou ohraničeny čerchovaně, jsou komponenty aplikace, které bylo nutné doimplementovat pro podporu práce s poznámkami a PDF dokumentem. Tyto prvky budou níže popsány podrobněji.



Obrázek 4.1: Diagram aktivit a dialogů aplikace.

Výběr dokumentu

Po spuštění aplikace PDF Annotation je aktivována aktivita **Choose File**, pomocí které uživatel vybere PDF dokument ke čtení či úpravě poznámek. Přístupem k menu aktivity může být pomocí položky *About* spuštěna aktivita **About**, která je implementována jako jednoduchý prohlížeč HTML stránky obsahující ovládání aplikace a licenční ujednání. Položka *Options* spustí aktivitu **Options**, která ze strukturovaného XML souboru načte veškeré nastavení aplikace, jež může uživatel změnit. Pomocí položky *Set folder as home* je možné nastavit aktuální adresář jako domovský.

Práce s dokumentem

Po vybrání dokumentu, který má být načten, je spuštěna aktivita **Open File**. Pokud uživatel přistoupí k hlavnímu menu aplikace, má na výběr mimo již dříve popsaných položek také položku *Go to page...*, která vyvolá dialog pro zadání konkrétního čísla stránky, na kterou je možné přejít. Položka *Find...* zobrazí dialog pro zadání hledaného výrazu, *Open...* znovu spustí aktivitu **Pages View** pro nový výběr PDF dokumentu a položky *Rotate left* a *Rotate right* slouží k otočení stránky o 90° ve směru nebo proti směru hodinových ručiček.

Seznam poznámek

Pomocí položky *Show Annotation List* hlavního menu aplikace je možné spustit aktivitu *Annotation List*, která zobrazuje seznam všech poznámek nalezených v PDF dokumentu. Vzhled seznamu je opět definován v samostatném XML souboru, obsah je dynamicky generován ze záznamů poznámek nacházejících se v databázi. Kromě obvyklých informací o poznámce je zde navíc obsažena informace o číslu stránky, na které se konkrétní poznámka nachází. Při doteku obrazovky na místě zobrazení konkrétního záznamu s poznámkou je pomocí metody `onClick` třídy `AnnotationList` vyvolán dialog obsahující celý záznam poznámky. Pomocí tohoto dialogu je možné přistoupit k úpravě obsahu konkrétní poznámky.

Vložení a úprava poznámky

Vložení nové poznámky má na starosti dialog `Insert Annotation`, který je stejně jako dialog `Annotation Contents` pro zobrazení obsahu a možnost úpravy poznámky, vyvolán událostí *Long Press*. Tato událost nastane při déle trvajícím doteku obrazovky.

Dialog `Insert Annotation` obsahuje tři položky pro výběr z konkrétního typu poznámky. Položky jsou implementovány pomocí privátní třídy `Item`, která umožňuje vytvořit položku reprezentovanou nejen textovým řetězcem, ale také zvolenou bitmapovou ikonkou. Po zvolení jedné z položek je spuštěna aktivita `Annotation`, pomocí které je možné vložit novou poznámku.

Implementace dialogu `Annotation Contents` je obdobná s tím rozdílem, že obsahuje položky čtyři. V případě volby položky pro úpravu poznámky je opět spuštěna aktivita `Annotation` s již předvyplněným obsahem poznámky načteným z konkrétního databázového záznamu. Volba pro smazání poznámky přepíše atribut *flag* v databázovém záznamu na hodnotu odpovídající smazané poznámce a upravovanou poznámku zneviditelní. Následují volby pro změnu pozice a změnu velikosti poznámky, které uzavřou tento dialog a k upravované poznámce vykreslí bitmapovou ikonku znázorňující zvolenou akci.

Změna polohy a velikosti poznámky

Při volbě změny polohy poznámky je poznámka opatřena bitmapovou ikonkou znázorňující přesun. Pomocí metody `onTouch` je pak detekována událost, kterou uživatel pomocí dotyku obrazovky vyvolá. Událost `ACTION_MOVE`, vyvolaná posunem, mění souřadnice poznámky dle místa dotyku a v reálném čase vykresluje aktuální polohu. V případě, že se uživatel dotkne vně poznámky, aktuální poloha je potvrzena a uložena.

Volba pro změnu velikosti poznámky nabízená u oválných a čtvercových poznámek vykreslí bitmapovou ikonku znázorňující změnu velikosti v pravém horním rohu poznámky. Podobně jako u změny polohy je i tady detekována událost, kterou uživatel svým dotekem obrazovky vyvolá, a v reálném čase překreslována velikost dle polohy dotyku obrazovky uživatele. Změna velikosti je možná dle svíslé i vodorovné osy. Po uložení dotykem vně poznámky jsou souřadnice přepočítány tak, aby rozdíl souřadnic levého dolního a pravého horního rohu nebyl záporný. Uživatel je o úspěšném uložení veškerých změn informován informačním hlášením.

4.5 Texty aplikace

Veškeré texty použité v aplikaci jsou stejně jako vzhled aktivit vytvořeny deklarativně v samostatném souboru jazyka XML. Tato skutečnost umožňuje velmi jednoduše nejen texty

upravovat, ale případně i vkládat nové v jiném národním jazyce. Každý textový řetězec je uzavřen do jednoho párového elementu `<string>` s povinným parametrem `name`, který slouží jako jedinečný identifikátor. V případě dialogu `Insert Annotation` nebo `Select action`, které nabízejí výběr z více položek, je použit párový element `<string-array>`, který obsahuje elementy `<item>` obsahující texty jednotlivých položek.

4.6 Uložení PDF dokumentu

Položka `Save as...` hlavního menu aplikace vyvolá dialog pro zadání názvu souboru a následné uložení, které je implementováno pomocí metody `saveAnnotToFile` ve třídě `Annotation` s využitím knihovny `iText`. Metoda provádí také kontrolu, zda soubor s tímto názvem už existuje. Pokud existuje, je vyvolán další jednoduchý dialog informující o této skutečnosti uživatele, který tuto akci potvrdí, či zamítne. Při potvrzení přepsání souboru či vytváření nového souboru je pomocí metody `confirmedSaveToFile` přepsán případně vytvořen nový soubor.

Nejdříve jsou načteny všechny záznamy s poznámkami z SQLite databáze a poté jsou podle hodnoty atributu `flag` jednotlivých záznamů prováděny příslušné operace s poznámkou. K provádění jednotlivých operací, mezi které patří vložení nové poznámky, úprava nebo smazání stávající poznámky PDF dokumentu, je použita instance třídy `PdfReader`. Pomocí této instance lze přistoupit ke kterémukoliv objektu v PDF dokumentu.

Vytvoření PDF objektu

V případě vložení nové poznámky je pomocí instance třídy `PDFStamper` vytvořen nový PDF objekt s touto poznámkou. K vytvoření objektu je třeba stanovit všechny PDF klíče, kterými se zabývá sekce 3.5.1. Nejprve je potřeba určit typ PDF objektu, kterým je v tomto případě poznámka a jehož hodnotou je klíč `/Annot`. Pozici poznámky reprezentuje pole se souřadnicemi dvěma bodů. Body stanovují horní pravý roh a dolní levý roh poznámky v souřadných osách a jsou určeny datovým typem `float`. Textová část poznámky, tedy jméno autora, předmět a samotný obsah poznámky, je zastoupena textovým řetězcem, který bylo nutné překódovat na znaky v kódování Unicode tak, aby poznámky uložené v PDF dokumentu obsahovaly korektní textovou informaci s diakritikou. Datum poslední změny poznámky je v databázovém záznamu uloženo ve tvaru, který není třeba dále upravovat.

Dále je potřeba rozlišit, o který podtyp PDF objektu, tedy poznámky, se jedná. V případě textové poznámky je nutné vytvořit klíč `/Text` a stanovit RGB barvu pozadí ikonky poznámky. Barva je v PDF dokumentu uložena jako tříprvkové pole, každý prvek pole reprezentuje jeden barevný kanál. V databázovém záznamu poznámky je barva uložena v hexadecimálním tvaru, který je nutno převést do korektního tvaru. V případě oválné a čtvercové poznámky je vytvořen klíč `/Circle` nebo `/Square`. U těchto poznámek je důležité určit barvu a šířku rámečku. Barva rámečku je stejně jako u předchozího typu poznámky reprezentována tříprvkovým polem klíče `/C`. Pro vytvoření rámečku je potřeba stanovit klíč `/Border` a vyplnit třetí prvek jeho pole, který nese informaci o šířce rámečku poznámky v podobě celého čísla.

Úprava existujícího PDF objektu

Pro úpravu stávající poznámky je nezbytné vyhledat stránku a poté objekt s touto poznámkou v PDF dokumentu podle reference, která je uložena v databázovém záznamu. Poté,

co je objekt identifikován, proběhne ověření, zda je nalezený objekt skutečně objektem poznámky. K tomuto účelu je opět použita instance třídy `PdfReader`. Úprava probíhá tak, že jsou všechny klíče objektu s poznámkou přepsány hodnotami, které nese daný databázový záznam poznámky.

Odstranění PDF objektu

Smazání existující poznámky probíhá přistoupením k objektu stránky PDF dokumentu, který obsahuje klíč `Annots` reprezentovaný polem referencí na PDF objekty poznámek dané stránky. V tomto poli je vyhledána reference na objekt konkrétní poznámky, která je načtena ze záznamu uloženého v databázi, a prvek pole s touto referencí z pole posléze odstraněn.

Po upravení objektů poznámek dokumentu prostřednictvím instance třídy `PdfReader` je modifikovaný dokument uložen pomocí metody `saveTo` do specifikovaného souboru. Aby mohla aplikace bez problému soubor uložit, vyžaduje povolení práv zápisu do souboru `WRITE_EXTERNAL_STORAGE`. Tato práva uživatel potvrzuje při instalaci aplikace. V případě, že i přes toto oprávnění dojde k chybě při ukládání souboru, je o této skutečnosti uživatel informován pomocí chybového hlášení objevujícího se na obrazovce.

Veškeré zdrojové kódy aplikace jsou dostupné on-line na serveru Google Code¹.

¹ Zdrojové kódy aplikace jsou volně přístupné na adrese: <http://code.google.com/p/pdf-annotation>.

Kapitola 5

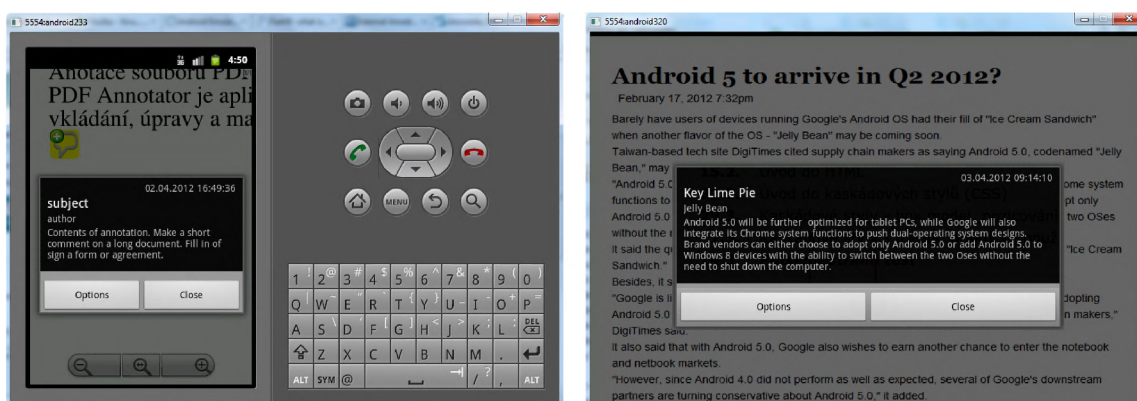
Testování

Jedním z nejdůležitějších bodů při vývoji aplikace pod platformou Android je její řádné otestování na různých verzích platformy. Testování bylo rozděleno do dvou etap. V první etapě byla aplikace testována v Android emulátoru, ve druhé etapě pak přímo na fyzických zařízeních. Testy byly zaměřeny zejména na stabilitu a spolehlivost aplikace včetně ověření intuitivity a správné funkčnosti uživatelského rozhraní.

5.1 Android emulátor

Prostředí Android SDK, které je zmíněno v sekci 2.2, nabízí Android emulátor mobilního zařízení, který je znázorněn na obrázku 5.1. Jde o virtuální mobilní zařízení, které lze spustit na běžném počítači např. v prostředí *Eclipse*, a které umožňuje vyvíjet a testovat Android aplikace bez použití fyzického zařízení [4].

V Android emulátoru byla aplikace testována na verzích platformy 2.1 *Eclair*, 2.2 *Froyo*, 2.3 *Gingerbread*, 3.0 *Honeycomb* a 4.0 *Ice cream sandwich* s různými velikostmi displejů. Důraz byl kladen na řádné otestování aplikace zejména na verzi 3.0, která je určena pro tablety s displejem větších rozměrů. Displeje mobilních telefonů, pro které byla aplikace primárně vyvíjena, disponují menšími rozměry a nižším rozlišením a bylo tedy potřeba upravit uživatelské rozhraní aplikace tak, aby bylo použitelné i pro uživatele tabletů.



Obrázek 5.1: Ukázka běžící aplikace v Android SDK emulátoru. Napravo je znázorněn emulátor mobilního telefonu, nalevo pak emulátor tabletu.

5.2 Fyzická zařízení

Aplikace byla ve druhé fázi testována na různých fyzických zařízeních s různými verzemi platformy Android, různými velikostmi displejů a různými frekvencemi procesorů tak, aby byla ověřena funkčnost aplikace z nejrůznějších hledisek.

Testování bylo prováděno na mobilním telefonu *Sony Ericsson Xperia X10i*¹, na kterém byla ověřována především rychlost vykreslování PDF dokumentu a ověřování práv pro čtení a zápis souborů nejen do paměti telefonu, ale i na vyměnitelné úložiště. Rovněž byl kladen důraz na intuitivnost a správné vykreslování uživatelského rozhraní. Telefon ověřil funkčnost aplikace na platformě Android verze 2.3. Druhým testovaným telefonem se stal *ZTE Blade*², na kterém byla ověřena plynulost aplikace na zařízeních s nižší frekvencí procesu a správná funkčnost na platformě Android verze 2.2. V pořadí třetím testovaným telefonem byl *Samsung S5570i Galaxy Mini*³, jehož úkolem bylo ověření chování aplikace na zařízeních disponujících nízkým rozlišením displeje.

Během testování aplikace se objevily problémy se správným zobrazením uživatelského rozhraní na zařízeních s vyšším rozlišením displeje a problémy s omezenými právy pro zápis souboru do určeného úložiště. Při testování v Android emulátoru a na fyzických zařízeních s nižší frekvencí procesoru byla zaznamenána delší odezva aplikace, a to zejména při načítání PDF dokumentu s větším množstvím poznámek.

Závislost verze PDF specifikace a počtu poznámek na načítání

Protože existuje více verzí specifikace PDF dokumentů uvedených v sekci 2.5 v tabulce 2.2, bylo nutné otestovat chování aplikace u každé z nich. Jako testovací zařízení byl zvolen mobilní telefon *Sony Ericsson Xperia X10i*, na kterém byl testován dvoustránkový dokument obsahující celkem 7 různých typů poznámek v různé verzi PDF specifikace. Dobu načtení testovacího dokumentu v aplikaci znázorňuje tabulka 5.1. Hodnoty v tabulce jsou průměrem z celkem deseti provedených měření. Nejvyšší hodnota 1 405,3 ms byla naměřena u specifikace PDF 1.6, naopak nejnižší hodnota 533,2 ms u specifikace PDF 1.7 EL 3. Z těchto hodnot lze vyčíst, že verze PDF specifikace nemá na načítání PDF dokumentů výrazný vliv.

Verze	PDF 1.3	PDF 1.4	PDF 1.5	PDF 1.6	PDF 1.7	PDF 1.7 EL 3	PDF 1.7 EL 8
Doba (ms)	663,72	637,15	608,62	774,79	679,41	687,67	674,74

Tabulka 5.1: Doba načítání závislá na verzi PDF specifikace.

Výraznější vliv na dobu načítání PDF dokumentů má ovšem množství poznámek. Podle testů bylo zjištěno, že za vyšší dobu načítání může pomalé ukládání poznámek do SQLite databáze, které se děje při každém načtení nového PDF dokumentu. Během optimalizace aplikace byl redukován počet SQL dotazů a tím dosaženo až 5x rychlejšího načítání PDF dokumentů, nicméně problém i nadále přetrvává. Pro testování byly vytvořeny PDF dokumenty s verzí specifikace 1.3 a 1.7 Extension Level 8. První dokument neobsahoval žádnou

¹Sony Ericsson Xperia X10i: mobilní telefon disponující platformou Android ve verzi 2.3.3 Gingerbread, velikostí displeje 4 palce a rozlišením displeje 854x480 bodů. Telefon je vybaven procesorem o frekvenci 1 GHz.

²ZTE Blade: mobilní telefon vybaven systémem Android verze 2.2 Froyo s velikostí displeje 3.5 palce, rozlišením displeje 800x480 bodů a frekvencí procesoru 600 MHz.

³Samsung S5570i Galaxy Mini: mobilní telefon pracující se systémem Android verze 2.3.4 Gingerbread. Telefon disponuje displejem o velikosti 3,14 palce a rozlišením 320x240 bodů. Procesor telefon pracuje na frekvenci 832 MHz.

poznámku, druhý jich obsahoval 10 a třetí 25 poznámek na pěti stranách. Z tabulky 5.2 je patrné, že při načítání PDF dokumentu s větším množstvím poznámek doba načítání lineárně roste. Měření také ukazuje závislost doby načítání na zvolené verzi PDF specifikace. Naměřené hodnoty jsou opět průměrem z celkem deseti provedených měření. Nejvyšší hodnota 1 631,8 ms byla naměřena u PDF dokumentu se specifikací PDF 1.3, který obsahoval 25 poznámek. Nejnižší hodnota 273,1 ms byla naměřena u dokumentu se stejnou specifikací, který neobsahoval žádnou poznámku.

PDF verze	PDF 1.3			PDF 1.7 Extension Level 8		
Počet poznámek	žádná	10	25	žádná	10	25
Doba vykreslení	329,76 ms	689,12 ms	1 528,17 ms	347,42 ms	611,16 ms	1 369,21 ms

Tabulka 5.2: Rychlost načítání PDF dokumentů v závislosti na počtu poznámek.

5.3 PDF dokument

Testován byl také vytvořený PDF dokument v různých verzích PDF specifikace. Na zobrazení a práci s poznámkami v aplikaci neměla verze specifikace žádný vliv. Při zobrazení na běžném počítači v prohlížeči PDF dokumentů Adobe Reader také nebyl vliv specifikace zaznamenán. Jediným rozdílem je různá grafická reprezentace textových poznámek v různých verzích tohoto prohlížeče. Adobe Reader neumožňuje upravovat oválné a čtvercové poznámky vytvořené v aplikaci. Pro úpravu těchto poznámek je třeba použít např. nástroj Adobe Acrobat.

5.4 Publikování aplikace

Poté, co byly opraveny veškeré nalezené chyby během testování, byla aplikace publikována na digitálním úložišti aplikací Google Play⁴, kterým se zabývá sekce 2.4. Díky publikování se aplikace může dostat k širokému spektru uživatelů z celého světa, od kterých je možné získat zpětnou vazbu. Publikování mezi uživatele je jediným zaručeným způsobem, jak zjistit, zda aplikace funguje opravdu správně a zda je pro uživatele užitečná a použitelná.

Rozšíření navržená uživateli

Uživatelé, kteří aplikaci testovali, navrhli několik dalších možných rozšíření. Mezi plánované rozšíření mimo jiné patří:

- Import a export poznámek do souboru FDF (Forms Data Format), který byl pro tento účel vyvinutý přímo společností Adobe.
- Možnost zvolit průhlednost textových poznámek tak, aby bylo možné číst text, který poznámky překrývají.
- Možnost předdefinovat vlastní poznámky s vlastním vzhledem a textovým obsahem.
- Implementace akcí Zpět/Vpřed při práci s poznámkami.

⁴ Aplikace je dostupná na adrese <https://play.google.com/store/apps/details?id=cx.pdf.android.pdfview>.

Kapitola 6

Závěr

Nejdůležitějším cílem této práce bylo navrhnout a implementovat aplikaci, která umožňuje zobrazení obsahu PDF dokumentu a pohodlnou práci s PDF poznámkami pro platformu Android. Dokument, který byl aplikací upraven, lze bez potíží zobrazit v běžném programu pro prohlížení PDF dokumentů na stolním počítači.

6.1 Přínos práce

Veškeré stanovené cíle této práce byly splněny. Podařilo se vytvořit funkční aplikaci, kterou ocení zejména čtenáři PDF dokumentů či elektronických knih, kterým chyběla možnost do PDF dokumentu vložit poznámku umožňující např. vysvětlení označeného slova či pojmu nebo vložení nové informace, která v PDF dokumentu chybí. Díky publikování na digitálním úložišti Google Play je aplikace dostupná pro drtivou většinu uživatelů, od kterých je možné sbírat nové návrhy na zlepšení. Aplikaci si stáhne průměrně 8 – 15 uživatelů denně.

6.2 Budoucnost projektu

Aplikace byla navrhována a implementována tak, aby uživateli nabídla možnost práce s nej-používanějšími typy poznámek v PDF dokumentu. Možné rozšíření aplikace se tedy nabízí o další poznámky, kterých existuje přibližně 26 typů.

Za další možné rozšíření by se dal považovat fakt, že poznámky nejsou vykreslovány pomocí nativního kódu *MuPDF* knihovny, kterou aplikace využívá. Pokud by se tedy tato knihovna rozšířila o možnost vykreslování všech druhů poznámek na úrovni nativního kódu, aplikace by zcela jistě dosáhla vyšší rychlosti a lepší odezvy.

Vývoj aplikace bude i nadále pokračovat. Každá nová verze bude publikována na digitálním úložišti Google Play a na základě zpětné vazby uživatelů průběžně opravovány nalezené chyby a nedostatky. Cílem je implementace podpory PDF objektů, pomocí kterých je možné určit přesný vzhled poznámek, nebo zvýraznit, podtrhnout či přeškrtnout označený text, ale také podpora více národních jazyků aplikace.

Literatura

- [1] Historie formátu PDF společnosti Adobe. *Adobe* [online]. 2012 [cit. 2012-03-03]. Dostupné z: <http://www.adobe.com/cz/products/acrobat/adobe.pdf.html>
- [2] Portable Document Format (PDF). *Adobe* [online]. 2012 [cit. 2012-03-03]. Dostupné z: <http://www.adobe.com/pdf>
- [3] Activities. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://developer.android.com/guide/topics/fundamentals/activities.html>
- [4] Android Emulator. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://developer.android.com/guide/developing/tools/emulator.html>
- [5] Application Fundamentals. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://developer.android.com/guide/topics/fundamentals.html>
- [6] Introducing Google Play. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://android-developers.blogspot.com/2012/03/introducing-google-play.html>
- [7] New Seller Countries in Google Play. *Android Developers* [online]. 2012 [cit. 2012-04-12]. Dostupné z: <http://android-developers.blogspot.com/2012/04/new-seller-countries-in-google-play.html>
- [8] Platform Versions. *Android Developers* [online]. 2012 [cit. 2012-02-02]. Dostupné z: <http://developer.android.com/resources/dashboard/platform-versions.html>
- [9] Publishing on Google Play. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://developer.android.com/guide/publishing/publishing.html>
- [10] Tasks and Back Stack. *Android Developers* [online]. 2012 [cit. 2012-02-02]. Dostupné z: <http://developer.android.com/guide/topics/fundamentals/tasks-and-back-stack.html>
- [11] User Interface. *Android Developers* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://developer.android.com/guide/topics/ui/index.html>
- [12] What is Android. *Android Developers* [online]. 2012 [cit. 2012-02-02]. Dostupné z: <http://developer.android.com/guide/basics/what-is-android.html>
- [13] FRIESEN, Jeff. *Learn Java for Android Development*. New York: Apress, 2010. ISBN 978-1-4302-3156-1.

- [14] APV PDF Viewer. *Google Code* [online]. 2012 [cit. 2012-01-28]. Dostupné z: <http://code.google.com/p/apv/>
- [15] Android and Security. *Google Mobile* [online]. 2012 [cit. 2012-03-12]. Dostupné z: <http://googlemobile.blogspot.com/2012/02/android-and-security.html>
- [16] qPDF Notes Demo. *Google Play* [online]. 2012 [cit. 2012-02-07]. Dostupné z: <https://play.google.com/store/apps/details?id=com.qoppa.activities.notes>
- [17] JACKSON, Wallace. *Android Apps for Absolute Beginners*. New York: Apress, 2011. ISBN 978-1-4302-3446-3.
- [18] LOWAGIE, Bruno. *iText in Action Second Edition: Creating and manipulating PDF*. Stamford: Manning Publications, 2011. ISBN 1932394796.
- [19] A lightweight PDF and XPS viewer. *MuPDF* [online]. 2011 [cit. 2011-10-20]. Dostupné z: <http://www.mupdf.com>
- [20] What is OpenJPEG. *OpenJPEG* [online]. 2011 [cit. 2011-10-20]. Dostupné z: <http://www.openjpeg.org>
- [21] The history of PDF. *Prepressure* [online]. 2012 [cit. 2012-03-03]. Dostupné z: <http://www.prepressure.com/pdf/basics/history>
- [22] Jbig2dec. *Sourceforge* [online]. 2011 [cit. 2011-10-20]. Dostupné z: <http://jbig2dec.sourceforge.net>
- [23] FreeType 2. *The FreeType Project* [online]. 2011 [cit. 2011-10-20]. Dostupné z: <http://www.freetype.org/freetype2/index.html>
- [24] The Android Story. *[X]Cube Labs* [online]. 2012 [cit. 2012-03-03]. Dostupné z: <http://www.xcubelabs.com/the-android-story.php>

Příloha A

Obsah CD

Na přiloženém CD se v adresáři `/xvejta01` nachází zdrojové soubory aplikace včetně technické zprávy a souboru `README.txt`.

Obsah adresáře na přiloženém CD:

```
/xvejta01
  /apk
    /bin – spustitelný instalační soubor aplikace .apk
    /doc – technická dokumentace vygenerovaná pomocí systému javaDoc
    /manual – návod k ovládání aplikace a demonstrační PDF dokument obsahující
              vzorové poznámky
    /src – zdrojové soubory aplikace
  /doc
    /pdf – technická zpráva ve formátu PDF
    /src – zdrojové soubory technické zprávy pro sazecí systém LATEX
  /README.txt – pokyny pro překlad aplikace v Android NDK a v prostředí Eclipse
```

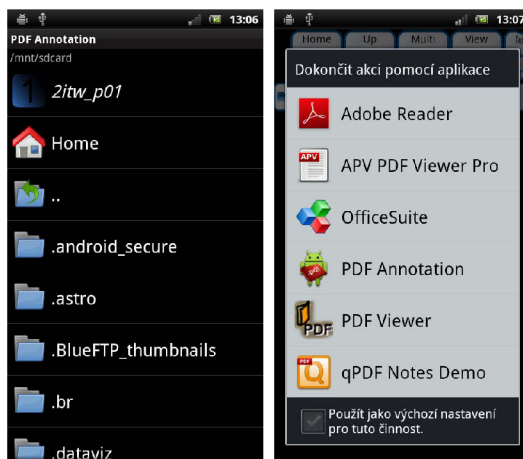
Příloha B

Uživatelská příručka

Návod na ovládání je nedílnou součástí každé aplikace. Tento návod se zabývá ovládáním aplikace PDF Annotation, která je založena na aplikaci APV PDF Viewer a umožňuje nejen PDF dokumenty prohlížet, ale také do těchto dokumentů vkládat své vlastní poznámky a libovolně s nimi manipulovat. Aplikace je určena pro mobilní zařízení s platformou Android verze 2.1 a vyšší.

B.1 Výběr dokumentu

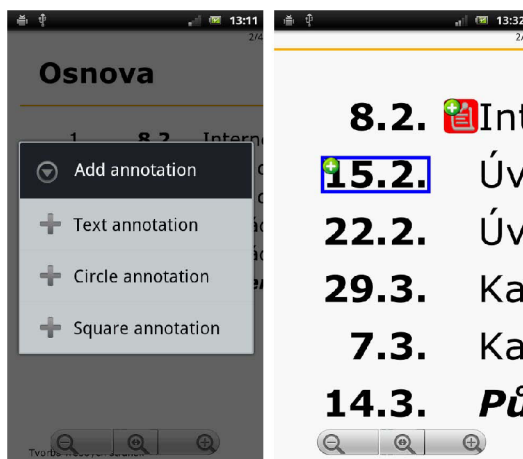
Po spuštění aplikace PDF Annotation je možné vybrat konkrétní PDF dokument, který chceme zobrazit. Slouží nám k tomu správce souborového systému, pomocí kterého můžeme procházet adresáře v mobilním zařízení. Druhou možností je libovolný software pro procházení adresářové struktury zařízení, s jehož pomocí lze vybrat PDF dokument pro prohlížení. V dialogu *Dokončit akci pomocí aplikace* pak stačí zvolit aplikaci PDF Annotation. Výběr dokumentu je znázorněn na obrázku B.1. V seznamu souborů se zobrazují naposled otevřené soubory, jejichž název je zobrazen kurzívou.



Obrázek B.1: Výběr dokumentu v aplikaci PDF Annotation a volba aplikace k otevření PDF dokumentu.

B.2 Vkládání poznámek

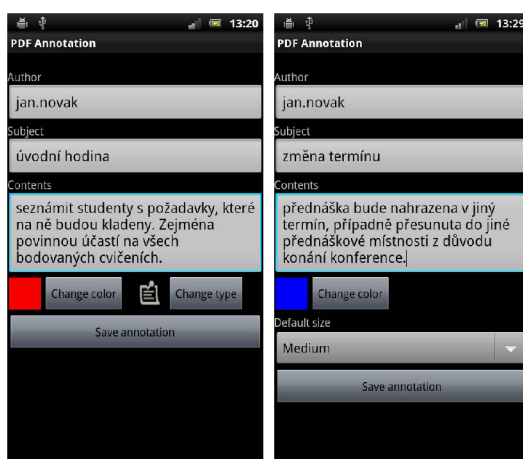
Po otevření vybraného PDF dokumentu do něj můžeme vložit libovolnou poznámku. Vložení poznámky provedeme pomocí dlouhého stisku v místě, kam chceme poznámku vložit. Pomocí dialogu, který se zobrazí po této akci, vybereme druh poznámky, který chceme vložit. Na výběr máme ze tří druhů poznámek: textová, oválná a čtvercová. Po vložení nové poznámky se u grafického znázornění poznámky objeví malá ikonka (plus v zeleném poli), která symbolizuje nově vloženou poznámku. Vkládání poznámek ilustruje obrázek B.2.



Obrázek B.2: Vložení poznámky a její grafická reprezentace.

B.2.1 Textová poznámka

Textová poznámka je v PDF dokumentu reprezentována pomocí grafické ikonky, jejíž vzhled i barvu pozadí lze zvolit. Samotný textový obsah poznámky se skládá ze jména autora, předmětu poznámky a vlastního obsahu poznámky. Vyplnění položky předmětu poznámky je povinné.



Obrázek B.3: Vložení nové poznámky: vlevo textové, vpravo oválné či čtvercové.

B.2.2 Oválná a čtvercová poznámka

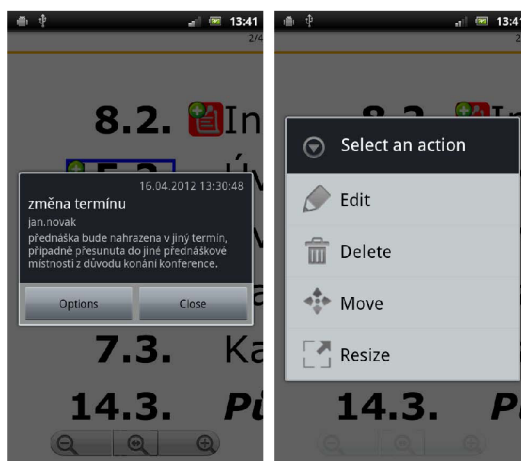
Oválná poznámka je v PDF dokumentu reprezentována pomocí vytvořeného oválu, čtvercová poznámka pomocí čtverce či obdélníku. U těchto poznámek je také možnost zvolit barvu rámečku a výchozí velikost vkládané poznámky. Vyplnění formuláře pro vložení nové poznámky je znázorněno na obrázku B.3.

B.3 Čtení a úprava poznámek

Možnost zobrazení obsahu poznámky je stejně jako možnost úpravy poznámky nedílnou součástí celé aplikace.

B.3.1 Čtení obsahu poznámky

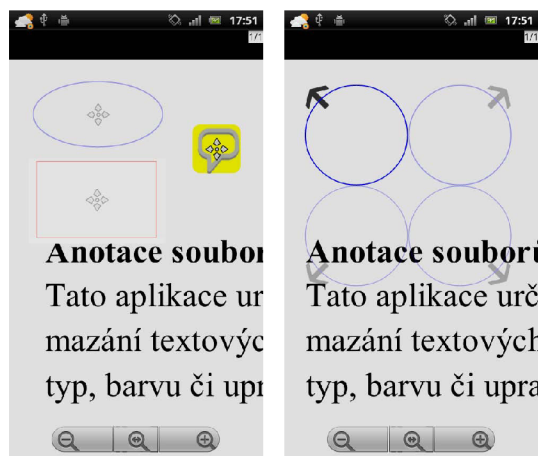
Přečtení obsahu poznámky docílíme pomocí dlouhého stisku v místě poznámky. Dialog, který se po této akci zobrazí, obsahuje informace o datu poslední změny poznámky, předmětu, autorovi a hlavním obsahu poznámky.



Obrázek B.4: Zobrazení obsahu poznámky a akce, které lze s poznámkou provádět.

B.3.2 Úprava poznámky

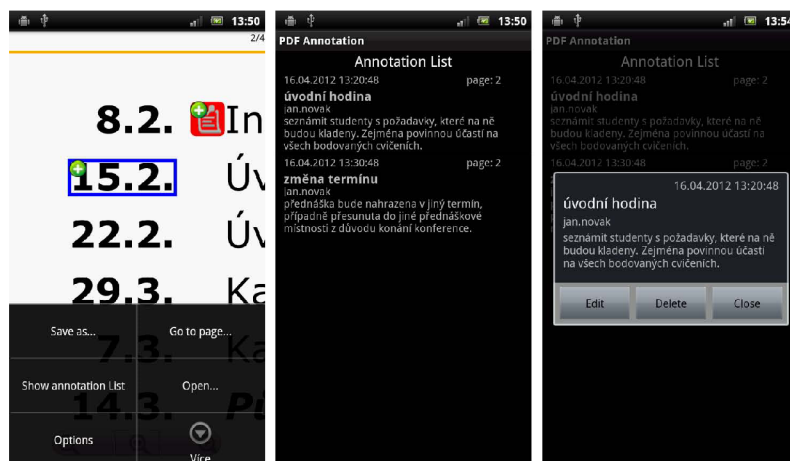
Po stisku tlačítka *Options*, které se nachází v dialogu s obsahem poznámky, máme možnost poznámku upravit. Tento dialog je znázorněn na obrázku B.4. Volba *Edit* slouží pro úpravu textové části poznámky, barvy pozadí ikonky či rámečku, případně pro změnu vzhledu ikonky. Pomocí volby *Delete* docílíme nenávratného smazání poznámky. V případě, že chceme poznámku přesunout na jiné místo v rámci stejné stránky PDF dokumentu, použijeme volbu *Move*. Pokud upravujeme oválnou či čtvercovou poznámku, v nabídce se objeví i volba *Resize*, která umožní změnit velikost poznámky. Po úpravě poznámky, která byla v PDF dokumentu vložena před jeho otevřením, se u grafického znázornění poznámky objeví červená vlaječka znázorňující, že byla poznámka upravena. Poznámky, které byly v PDF dokumentu uloženy před jeho otevřením, jsou označeny modrou vlaječkou. Přesun a změna velikosti poznámky je ilustrován na obrázku B.5.



Obrázek B.5: Přesun a změna velikosti poznámek.

B.4 Seznam poznámek

V menu aplikace se nachází položka *Show annotation list*, která zobrazí seznam všech poznámek v otevřeném PDF dokumentu. Tento seznam zobrazuje datum poslední změny poznámky, číslo stránky, na které se poznámka nachází, předmět a autora poznámky a v neposlední řadě také hlavní obsah poznámky. Po kliknutí na položku poznámky je otevřen dialog s podrobným popisem poznámky. Tento dialog obsahuje tlačítko *Edit* pro možnost přímého upravení poznámky a tlačítko *Delete* pro nenávratné smazání. Práce se seznamem poznámek je ilustrována na obrázku B.6.



Obrázek B.6: Ukázka práce se seznamem poznámek.

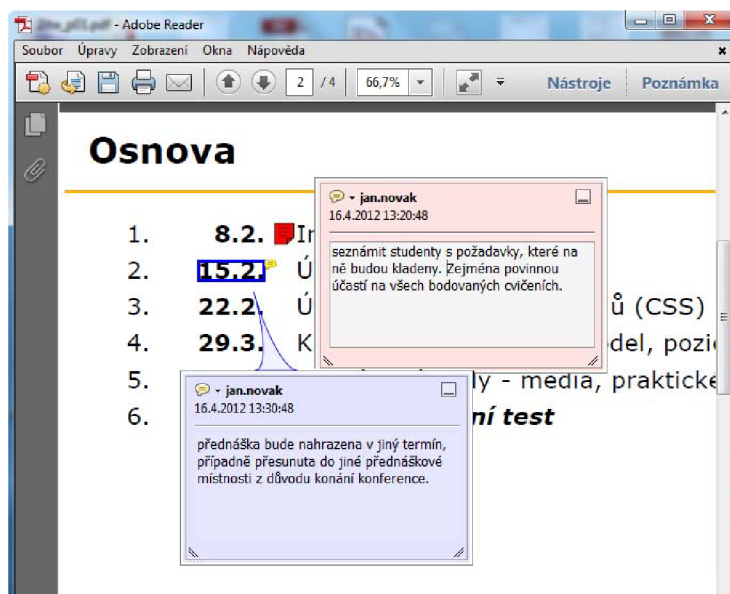
B.5 Uložení dokumentu

Před zavřením každého upraveného PDF dokumentu aplikace uživateli vždy nabídne uložení provedených změn. Uložení dokumentu je také možné z menu aplikace pomocí volby *Save as*.

Pokud by došlo k přepsání existujícího dokumentu, uživatel je o této skutečnosti informován.

B.6 Zobrazení dokumentu v počítači

Poznámky vytvořené pomocí aplikace PDF Annotation lze zobrazit v i běžném počítači. Mezi nejpoužívanější programy pro zobrazování PDF dokumentů patří např. Adobe Reader. Zobrazení poznámek v tomto programu je znázorněno na obrázku B.7. Tento program bohužel umožňuje upravovat pouze textové poznámky.



Obrázek B.7: Ukázka zobrazení poznámek v programu Adobe Reader X.