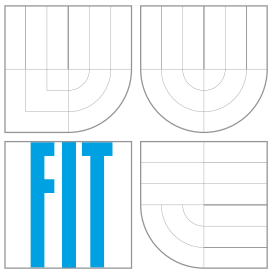
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

SPRÁVCE PREZENTACÍ PRO ANDROID

PRESENTATION MANAGER FOR ANDROID

BAKALÁRSKA PRÁCA
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MATÚŠ VANČO

VEDÚCI PRÁCE
SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2012

Abstrakt

Bakalárska práca sa zaoberá mobilnou platformou Android, a popisuje návrh a implementáciu aplikácie pre správu prezentácií pre túto platformu. Aplikácia umožňuje organizáciu blokov prezentácií a poskytuje pomoc pri prezentáciách. Dôraz je kladený na využívanie novátorských princípov platformy Android.

Abstract

This bachelor thesis deals with the Android mobile platform and describes design and implementation of application for managing presentations. Application allows you to organize blocks and assist in presentations. The emphasis is on using innovative principles of the Android platform.

Kľúčové slová

Android, prezentácia, mobilné zariadenie, aplikácia, Java, Google, organizácia, SQLite

Keywords

Android, presentation, mobile device, application, Java, Google, organization, SQLite

Citácia

Matúš Vančo: Správce prezentací pro Android, bakalárska práca, Brno, FIT VUT v Brně, 2012

Správce prezentací pro Android

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Zbyňka Křivku.

.....
Matúš Vančo
16. května 2012

Poďakovanie

Rád by som chcel poďakovať Ing. Zbyňkovi Křivkovi, vedúcemu bakalárskej práce, za odbornú pomoc a ochotu pri konzultáciách, a motiváciu, vďaka ktorej som túto prácu dokončil.

© Matúš Vančo, 2012.
Táto práca vznikla ako školské dielo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práca je chránena autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou zákonom definovaných prípadov.

Obsah

Kapitola 1

Úvod

Prezentácia alebo obhajoba práce je nevyhnutnou súčasťou každého väčšieho projektu, či uz sa jedná o školský projekt, alebo reálny projekt vo firme. Od nej veľakrát závisí či bude projekt správne ohodnotený a či zaujme cieľovú klientelu. Takisto aj organizácia výberového konania, alebo bloku prezentácií v školských laviciach by nemala byť podceňovaná. Pre správny výber a ohodnotenie kvalitných prác je potrebné presne dodržiať časový harmonogram a pružne reagovať na možné problémy, ktoré pri organizácii vznikajú. Mobilné zariadenia nám pri týchto situáciách môžu pomôcť zvýšiť kvalitu prezentácie a zefektívniť organizáciu.

Táto dokumentácia popisuje celý priebeh riešenia projektu *Správce prezentací pro Android*. Cieľom tohto projektu je vytvoriť elektronického správcu prezentácií pre mobilné zariadenia s dotykovým displejom. Program má pracovať pod inovatívnym operačným systémom Android. Pri riešení je kladený veľký dôraz na využívanie novátorských princípov platformy Android a hlavne vysokú ergonómiu dotykového užívateľského rozhrania, ktorá má významne urýchliť prácu s programom.

Prvá kapitola, *Základné koncepty Androidu*, zoznamuje čitateľa so základnými piliermi operačného systému Android. V kapitole *Špecifikácia a analýza požiadavkov* sú zhrnuté a vhodne rozšírené požiadavky na fungovanie aplikácie tak, aby aplikácia využívala možnosti operačného systému Android. Ďalšia kapitola *Návrh a implementácia aplikácie* rozdeľuje riešenie celej aplikácie na podproblémy, určuje kompetencie každého jedného z nich, navrhuje konceptuálne riešenia daných problémov a popisuje priebeh implementácie projektu pre operačný systém Android. V kapitole *Testovanie* je popísané, ako bola implementovaná aplikácia testovaná a bude v nej zhodnotená praktická použiteľnosť. Posledná kapitola *Záver* zhrňuje cieľ práce, ukazuje jej prínos a zoznamuje čitateľa s možnosťami ďalšieho vývoja aplikácie v budúcnosti.

Využitie mobilných zariadení pri správe prezentácií

Mobilné zariadenia môžu pri prezentácii pomáhať tak, že budú našeptávať prezentujúcemu na akom snímku by sa mal nachádzať, aké kľúčové informácie by mal pri aktuálnom snímku spomenúť alebo kedy treba zrýchliť priebeh prezentácie. Môžu však uľahčiť a zefektívniť aj prípravu na prezentáciu a poskytnúť prezentujúcemu prostriedky pre nácvik prejavu. V zariadeniach sa dá využiť diskrétne vibrovanie alebo zvuková signalizácia na upozornenie na rôzne udalosti alebo mikrofón na záznam zvuku pre tieto účely.

Pri organizácii blokov prezentácií sa dajú mobilné zariadenia tiež s výhodou použiť.

Môžu zobraziť organizátorom kedy začne nasledujúca prezentácia, kto budú jej autori. Môže upozorňovať na blížiaci sa koniec prezentácie. V prípade nečakaných situácií môže aplikácia pružne preusporiadať časy prezentácií a urýchliť tak priebeh celej akcie.

Kapitola 2

Základné koncepty platformy Android

Android je operačný systém vyvíjaný spoločnosťou *Open Handset Alliance*[16, 14], ktorý ako taký zahŕňa aj základné užívateľské programy vrátane emailového klienta, kalendáru, internetového prehliadača apod.[10]. Android umožňuje vyvíjať veľmi sofistikované aplikácie s využitím rozmanitého hardwaru dostupného na dnešných mobilných zariadeniach.

Operačný systém využíva Linuxové jadro, v ktorom je každá aplikácia spúšťaná pod unikátnym užívateľom. Každá aplikácia má nastavené oprávnenia iba pre konkrétneho užívateľa, ktorý danú aplikáciu spúšťa. Teda len užívateľ s ID priradením k aplikácii môže pristupovať k aplikačným súborom. Implicitne je dokonca každá aplikácia spúšťaná vo vlastnom procese. Tento proces sa spustí keď je spustený nejaký blok aplikácie a zruší sa, keď už nie je viacej potrebný, alebo musí systém uvoľniť novú pamäť iným aplikáciám. Tento odstavec a nasledujúce podkapitoly čerpajú informácie z knihy Marka Murphyho *Android 2*[12] a mnohých článkov na stránke *Android Developers* [3, 2, 15, 6].

2.1 Aplikačné bloky

Aplikácie sú v operačnom systéme Android rozdelené do aplikačných blokov (application components). Aplikačný blok predstavuje samostatnú časť aplikácie, ktorá môže byť nezávisle využitá. Veľkou výhodou takého prístupu je znovupoužiteľnosť celých aplikačných blokov v úplne iných aplikáciách. Napríklad môžeme využiť formulár na poslanie emailu z iného emailového klienta a užívateľovi sa zdá akoby bol tento formulár súčasťou našej aplikácie. Android rozlišuje štyri aplikačné bloky.

Aktivity

Aktivita (activity) je časť programu, ktorá má po celý čas definované jedno užívateľské rozhranie a má jednu úlohu. Napríklad v prípade emailového klienta môžeme chápať ako jednu aktivitu formulár na poslanie emailu. V Androide sú aktivity implementované pomocou triedy `Activity`.

Aplikácia pozostáva z viacerých aktivít, ktoré sú medzi sebou previazané. Jedna aktivita je označená ako hlavná (main) a líši sa od ostatných tým, že je zavolaná pri kliknutí na ikonku programu v spúšťači programov (launcher). Avšak do programu sa môžeme dostať aj prostretníctom inej aktivity tým, že druhá aplikácia priamo zavolá túto aktivitu.

Ak aktivita zavolá novú aktivitu, pozastaví sa a nová aktivita sa vloží na vrchol spätného zásobníku (back stack)[2] a presunie sa do popredia. Po tom ako užívateľ ukončí prácu s novou aktivitou a klikne na tlačítko „späť“, je nová aktivita vybraná zo zásobníku a spustí sa aktivita aktuálne ležiaca na vrchole zásobníku, teda aktivita, ktorá vyvolala zrušenú aktivitu.

Aktivita sa môže nachádzať v týchto stavoch:

- **bežiaca** (running) - je na popredí a dostupná užívateľovi na interakciu (je viditeľná užívateľovi a nachádza sa na vrchole zásobníka);
- **pozastavená** (paused) - užívateľ nemôže s aktivitou interagovať, ale beží a je viditeľná na obrazovke;
- **stopnutá** (stopped) - je úplne prekrytá inou aktivitou a teda nie je viditeľná na obrazovke, ale zachováva rôzne stavové informácie;
- **vypnutá** (destroyed) - pri presune na popredie sa aktivita odznova načíta a obnoví stav, ktorý programátor trvalo uložil pred vypnutím.

Obrázok 2.1 zobrazuje prehľadne životný cyklus aktivity. Šedé obdĺžniky reprezentujú *metódy spätného volania* (callback methods), ktoré implementujú v programe prechody medzi jednotlivými stavmi aktivity. Farebné ovály sú hlavné stavy, v ktorých sa môže aktivita nachádzať.

Na obrázku môžeme rozlíšiť 3 smyčky, ktoré sú pre život aktivity nevyhnutné. Každá zachytáva životný cyklus nejakej vlastnosti aktivity. Každá aktivita má životný cyklus:

- **existencie** - tento životný cyklus začína zavolaním funkcie `onCreate()`, kedy sa aktivita vytvorí a končí zavolaním funkcie `onDestroy()` pri rušení aktivity;
- **viditeľnosti** - životný cyklus začína zavolaním funkcie `onStart()`, kedy aktivita začne vykonávať svoju činnosť (užívateľ ale nemôže interagovať) a končí volaním funkcie `onStop()` (aplikácia prestane byť viditeľná);
- **interaktivity** - aktivita po zavolaní metódy `onResume()` sa presune do popredia a začne interagovať s užívateľom a po zavolaní metódy `onPause()` prestáva reagovať na príkazy používateľa.

Ďalšie informácie o aktivite a jej životnom cykle sú možné nájsť na oficiálnej stránke operačného systému Android určenej pre vývojárov[2].

Služby

Služba (service) je časť programu, ktorá nemá vlastné užívateľské rozhranie. Má jednu úlohu, avšak nie je možné túto úlohu parametrizovať priamou interakciou užívateľa. Neblokuje teda aktuálne bežiacu aktivitu. Táto funkcionalita je implementovaná v triede `Services`.

Poskytovatelia obsahu

Poskytovateľ obsahu (content provider) je časť programu, ktorá spravuje statické dáta aplikácie. Tieto dáta môžu byť využívané aj inými aplikáciami, pokiaľ je im to umožnené. Android umožňuje ukladanie perzistentných dát do jednoduchej SQLite databázy, na web, do súborového systému alebo na iné miesto, ku ktorej má aplikácia povolený prístup. Poskytovateľ obsahu je implementovaný v triede `ContentProvider`.

Príjmatelia všesmerových správ

Aplikačná komponenta *príjmateľ všesmerových správ* (broadcast receiver) reaguje na systémové všesmerové oznámenia. Veľa správ pochádza priamo zo systému. Napríklad systém oznamuje že bol vypnutý displej alebo je vybitá batéria. Ale aj samotné aplikácie môžu vyvolávať všesmerové správy. Príjmateľ všesmerových správ, hoci nemá definované užívateľské rozhranie, môže vytvoriť oznámenie v stavovom paneli upozornení, aby upozornil užívateľa. Príjmateľov všesmerových správ implementuje trieda `BroadcastReceiver`.

Prepojenie aplikačných blokov

Aplikačné bloky *aktivita*, *služba* a *príjmateľ všesmerových správ* sú aktivované prostredníctvom asynchronnej správy *Intent*. Správa Intent spája jeden komponent s druhým počas behu programu. Na vyvolanie konkrétnej *aktivity* alebo *služby* je potrebné definovať buď akciu, ktorá sa má zrealizovať (napríklad poslanie mailu) a systém vyberie spustenie komponenty, ktorá túto akciu realizuje, alebo môžeme priamo zadať, ktorá *aktivita* alebo *služba* sa má spustiť.

2.2 Špecifické komponenty aplikácie

Aplikácia obsahuje okrem zdrojového kódu v jazyku Java aj iné prostriedky pre robustnejšie a flexibilnejšie programovanie. Jedným z týchto prostriedkov je v súčastnosti veľmi často používaný formát XML na ukladanie statických prvkov aplikácie. Používa sa na transparentnejšie ukladanie návrhu grafického užívateľského rozhrania. Výhodou tohto prístupu je oddelenie aplikácie od zdrojov, ktoré sa môžu meniť v závislosti na konfigurácii zariadenia (rozlíšenie obrazovky, otočenie displeja, zmena jazyka aplikácie,...). Android ďalej poskytuje aj podporu pre databázové aplikácie. SQLite databáza je priamo súčasťou platformy Android.

Aplikačné prostriedky

Aplikačné prostriedky sú súbory pribalené k aplikácii a poskytújúce multimediálny obsah. Za prostriedky môžu byť považované obrázky, zvukové súbory, videá, ale aj XML súbory. XML súbory sú textové súbory, ktoré sa často používajú na uloženie rozličných štrukturovaných dát. Popis formátu XML súborov je pekne vysvetlený v knihe *Java a XML*[11] od Pavla Herouta. Všetky tieto súbory sa musia nachádzať v adresári `res/`, ktorý sa nachádza v inštalačnom súbore aplikácie.

V XML súboroch sa uchovávajú popisky aplikácie, ktoré sa pri lokalizácii aplikácie do iného jazyka dajú nahradiť a aplikácia pri ďalšom spustení funguje bez akýchkoľvek iných zásahov do jej súborov. XML súbory sú používané aj na rozvrhnutie grafických prvkov na obrazovke zariadenia.

Špecifickým typom zdrojov využívajúcich XML súbory sú Preferencie[5]. Preferenie popisujú grafické uživateľské rozhranie, ktoré je určené na konfiguráciu aplikácie. Rozdielom oproti bežným ostatným XML zdrojom je ten, že android pri použití Preferencie automaticky vytvára aplikačnú logiku na ukladanie globálnych nastavení.

Prostriedky[4] sú statickým prvkom aplikácie (rovnako ako je aj Java kód), ktorý sa využíva iba na čítanie. Teda napríklad obrázky, ktoré si vytvorí aplikácia v rámci svojej činnosti (použitím fotoaparátu atď.) nie sú aplikačné prostriedky. Hlavným dôvodom oddelenia prostriedkov od funkčného kódu je schopnosť poskytovať alternatívne prostriedky

pre odlišné konfigurácie zariadenia. Logika aplikácie ostáva zachovaná ale využíva pre svoj chod odlišné prostriedky pri odlišnom prostredí (odlišnom jazykovom nastavení, vybavení HW, natočení displeja atď.).

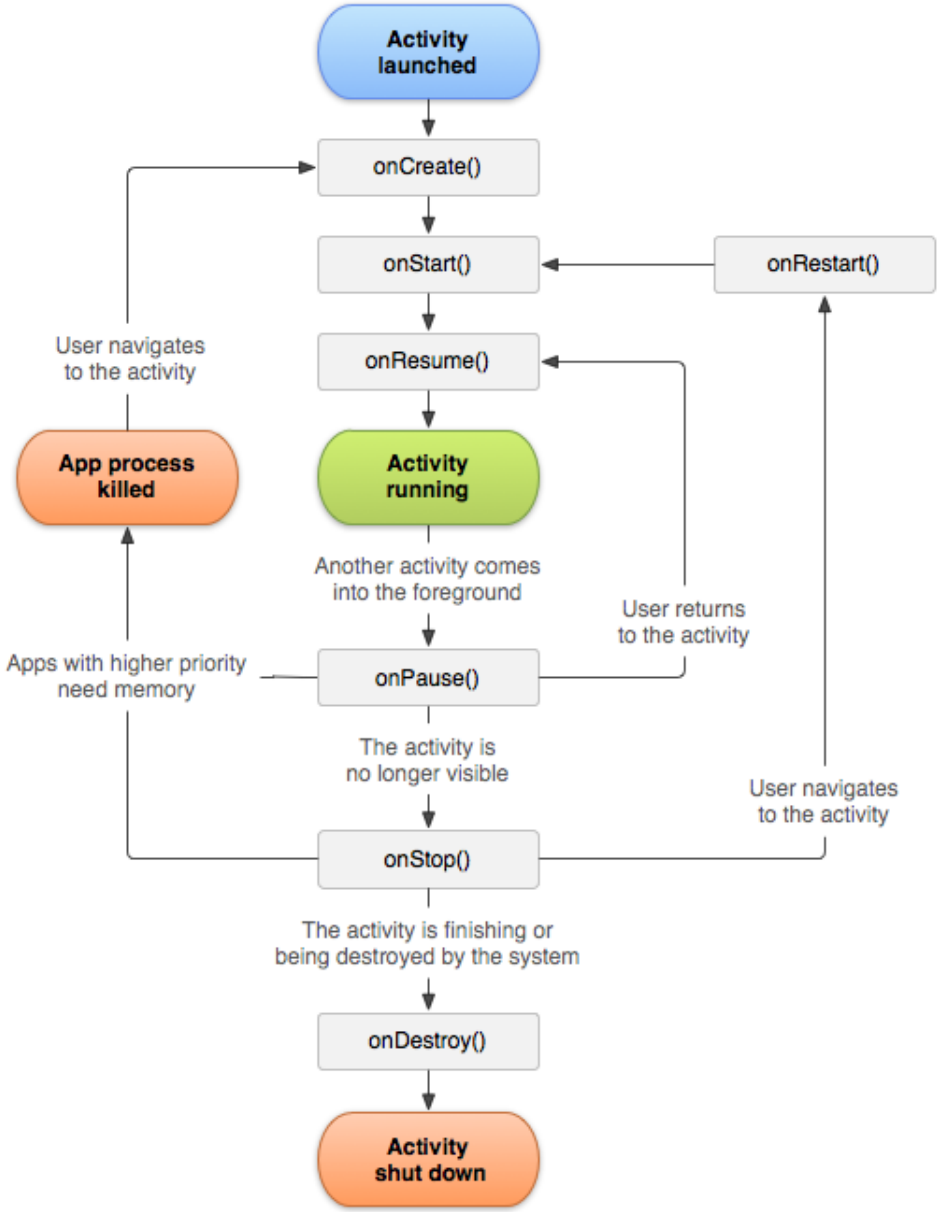
SQLite databáza

SQLite databáza je obľúbená databáza pre mobilné platformy, pretože obsahuje čisté SQL rozhranie, má veľmi malú pamäťovú náročnosť a je dostatočne rýchla. Naviac sa jedná o verejnú doménu, ktorú môže používať každý. Implicitne však neposkytuje niektoré vymoženosti väčších databáz. Neobsahuje typovú kontrolu stĺpcov (stĺpcom tabuľky môžeme síce priradiť typ INTEGER alebo TEXT, avšak to slúži len na interpretáciu uložených hodnôt, vložiť môžeme aj hodnoty iných typov) a ani integrigné obmedzenie typu *cudzí kľúč* (FOREIGN KEY). Tieto problémy sa dajú obísť prísnejšou aplikačnou logikou alebo ešte praktickejšie použitím *triggerov*. V systéme Android je databáza súčasťou jeho runtime prostredia, takže ju môže používať každá aplikácia.

Súbor manifestu

Základom akejkoľvek aplikácie pre Android je súbor manifestu, `AndroidManifest.xml`[8, 12], ktorý sa nachádza v koreňovom adresári v inštalačnom súbore. V ňom sa deklaruje obsah aplikácie. V manifeste sa deklaruje minimálna verzia Android SDK a prístupové práva, ktoré aplikácia pre svoj chod vyžaduje. Tieto požiadavky sú kvôli bezpečnosti zobrazené užívateľovi pri inštalácii aplikácie. V prípade že by aplikácia vyžadavala práva neprimerané k jej účelu, môže užívateľ zrušiť inštaláciu a uchrániť sa tak bezpečnostnej hrozbe a podozrivým aplikáciam. Ďalej sa tu deklarujú aplikačné bloky a ich triedy, ktoré sú zodpovedné za ich činnosť.

Prekladové nástroje systému Android automaticky generujú tento súbor iba pri vytvorení nového projektu. Pri každom pridaní nového aplikačného bloku alebo pri využití nového systémového prostriedku je treba pridať príslušnú deklaráciu do manifestu.



Obrázek 2.1: Životný cyklus aktivít[2]

Kapitola 3

Špecifikácia a analýza požiadavkov

Zadaním projektu je vytvorenie aplikácie, ktorá má fungovať ako elektronický správca prezentácií. Hlavným cieľom je organizácia časových blokov prezentácií a záznam poznámok počas nich. Aplikácia by mala zvládať tri módy činnosti – pre prezentácie, konferencie a obhajoby.

Aplikácia je určená pre mobilné zariadenia s operačným systemom Android. Je inšpirovaná priebehom obhajob na Fakulte informačných technológií Vysokého učení technického v Brně. Kľúčové atribúty návrhu sú vysoká ergonómia uživatelského rozhrania, flexibilita a dobré konfiguračné schopnosti.

Systém by mal organizovať časový harmonogram prezencácií, podávať informácie o aktuálnom dianí na prezentáciách a riešiť typické problémy prezentácií. Aplikácia by mala podporovať import a export poznámok a hodnotení vo formáte CSV. Toto by malo uľahčiť organizáciu prezentácií ale aj výrazné ušetriť čas.

3.1 Mód pre organizátora prezentácií

Tento mód bude poskytovať prostriedky pre plánovanie prezentácií, zobrazovanie informácií o aktuálne prebiehajúcej prezentácii a naplánovaných prezentáciách, umožňovať eliminovať časové prieťahy obhajob, importovať (exportovať) dáta do (z) systému a riešiť ďalšie typické problémy pri organizácii prezentácií. Tento mód bude teda zachycovať požiadavky ako pri obhajobách tak aj pri konferenčných prezentáciách.

Organizácia časového harmonogramu

Aplikácia bude umožňovať naplánovanie prezentácie na daný dátum a čas. Atribúty prezentácie sa budú dať v čase explicitne meniť v prípade neočakávaných situácií (časový posun kvôli omeškaniu). Zmeny môžu byť vyvolané aj na základe udalostí a upravené automaticky (zmena času jednej prezentácie tak že sa dve prezentácie prekrývajú v čase spôsobí automatický posun ďalšej prezentácie).

Pri plánovaní prezentácií sú dôležité atribúty okrem času a dátumu aj názov projektu, mená autorov projektu, mená organizátorov a maximálny čas obhajoby. Ďalej sa môžu nastavovať aj doplnkové parametre ako sú priorita alebo dodatočné poznámky k prezentácii.

Zobrazovanie informácií o aktuálnom dianí

Systém bude umožňovať prehliadanie naplánovaných prezentácií v dvoch režimoch. Jeden režim bude nezávislý na aktuálnom čase a bude zobrazovať obhajoby v chronologickom poradí (niečo ako prehliadanie záznamov v kalendári). Druhý bude zobrazovať aktuálne prebiehajúcu prezentáciu podľa aktuálneho času, teda vždy zobrazí v jednom čase a iba jednu obhajobu.

Prvý režim bude prehľadne informovať o aktuálne bežiacej prezentácii a o prezentáciach, ktoré najbližšie za ňou následujú. Pri každej prezentácii sa bude dať zistiť čas začiatku, trvanie, názov práce a názov prezentujúcich.

Druhý režim, kedže zobrazuje iba aktuálnu obhajobu, bude môcť prezentovať aj podrobnejšie informácie o obhajobe. Systém bude zobrazovať stopky a graficky znázorňovať časový priebeh aktuálnej obhajoby.

Riešenie typických problémov

Aplikácia bude okrem organizácie časového harmonogramu riešiť aj niektoré konkrétne problémy, ktoré nastávajú pri prezentáciách. Veľmi častým problémom je nedodržanie stanoveného maximálneho času na prezentáciu. Aplikácia v tomto prípade pomôže tak, že pred vypršaním časového limitu na prezentáciu upozorní užívateľa buď zvukovým signálom alebo vibrovaním zariadenia.

Import a export

Správca prezentacií bude umožňovať importovať a exportovať naplánované obhajoby. Import a export bude realizovaný prostredníctvom textového súboru v CSV formáte. Pri importe musia byť uvedené všetky povinné atribúty naplánovaných prezentácií, inak nebude import možný. Importovať sa budú aj všetky zadané voliteľné atribúty. Voliteľný atribút „čas začiatku prezentácie“ bude v prípade vynechania dopočítaný podľa ukončenia prezentácie uvedenej na predošlom riadku v CSV súbore. Export bude plne konfigurovateľný a bude teda umožnené užívateľovi vybrať si, ktoré stĺpce sa majú zobraziť vo výslednom súbore.

Rozšírenia požiadavkov

Systém môže umožňovať zvukový záznam obhajoby, ktorý môže slúžiť študentom, aby si po skončení obhajoby vypočuli chyby v ich prejave a tak sa im mohli v budúcnosti vyvarovať. Do zvukového záznamu sa môžu pridávať meta-informácie, ktoré by informovali poslucháča na akom snímku sa v danom čase prezentujúci nachádzal.

3.2 Mód pre prezentujúcich

Tento mód, určený pre prezentujúcich, bude umožňovať prezentujúcim sledovať stav svojej prezentácie z časového hľadiska a pomáhať aj pri samotnej prezentácii (týka sa najmä dodržania časového harmonogramu).

Pomoc pri obhajobe

Aplikácia bude zobrazovať užívateľovi počas prezentovania časový priebeh obhajoby a číslo snímku, na ktorom by sa mal nachádzať. Užívateľ bude môcť zadať pre presnejšiu kont-

rolu číslo snímku, na ktorom sa momentálne nachádza a aplikácia prepočíta čas vyhradený pre nasledujúce snímky. Tak bude môcť dohnať prípadný časový deficit v prezentácii. Pri aplikácii sa bude prehľadne zobrazovať aj množstvo času, ktoré má užívateľ na konkrétny snímok. Takisto sa budú pri každom prepnutí snímku aj ukazovať prezentujúcemu dodatočné poznámky ku snímku, ktoré si prezentujúci predtým pripravil.

Rozšírenie požiadavkov

Aplikácia bude pomáhať aj v príprave na samotnú prezentáciu. Aplikácia bude prezentujúcim umožňovať nahrávanie ich prejavu a meranie času potrebného na konkrétny snímok a na celú prezentáciu. Po ukončení merania bude aplikácia prepočítavať čas strávaný na snímkoch podľa optimálneho času na prezentáciu tak, aby bol zachovaný časový pomer medzi jednotlivými snímkami. Teda v prípade že bola prezentácia dlhšia ako plánovaný čas na ňu, program skráti časy strávené na snímku, pričom snímky, ktoré vyžadovali viacej priestoru budú zohľadnené a budú mať aj po prepočte dlhšie trvanie. Tento prepočet môže využiť samotná aplikácia na to, aby dávala užívateľovi veľmi presné informácie kedy má prepnúť ďalší snímok tak, aby bol dodržaný časový limit na prezentáciu. Pokiaľ si užívateľ nebude môcť dovoliť nahrávanie zvukových stôp (chýbajúci mikrofón alebo nedostatok pamäte na zariadení) alebo to nebude vyžadovať, bude si môcť zvoliť režim bez záznamov a v tom prípade sa bude iba merať čas potrebný na jednotlivé úseky prezentácie. Tieto informácie poslúžia ako spätná odozva pred samotnou prezentáciou a umožnia prezentujúcemu sa na zistené skutočnosti pripraviť.

3.3 Analýza požiadavkov

V tejto podkapitole bude porovnaná vyvíjaná aplikácia s ostatnými existujúcimi aplikáciami na Android Markete, ktoré majú podobné zameranie a slúžia na správu prezentácií.

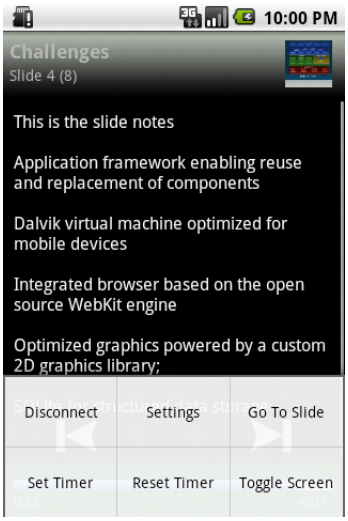
Existujúce aplikácie

Platforma Android sa nachádza na trhu dostatočne dlhú dobu a má širokú podporu programátorov po celom svete a preto existuje veľa aplikácií s podobnou funkcionalitou a zámerom. V súčasnosti je veľmi ťažké nájsť typ aplikácie, ktorý ešte nikto nevymyslel. Preto je vhodné sa pred písaním novej aplikácie zoznámiť so zaujímavými aplikáciami na webe[1] kvôli inšpirácii a odlíšeniu aplikácie od ostatných.

Remote for PowerPoint: Aplikácia umožňuje na diaľku ovládať prepínanie snímkov na PC. Pripája sa cez Bluetooth alebo WiFi. Zobrazuje snímky aj na mobilnom telefóne. Umožňuje nastaviť čas prezentácie a sledovať dobu, ktorá uplynula. Podporuje všetky verzie programu Microsoft PowerPoint, na vytváranie a prehrávanie prezentácií. Na obrázku 3.1 sa nachádza ukážka obrazovky tento aplikácie.

Presentation Timer: Jednoduchá aplikácia na nácvik prezentácií. Upozorňuje užívateľa na blížiaci sa koniec prezentácie zmenou farby pozadia okna. Pri spustený je nastavená farba na zelenú a postupne prechádza do žltej a následne do červenej farby.

Presentation Pal: Profesionálna aplikácia, ktorá nepotrebuje pri premietaní snímkov ani počítač. Dokáže sa priamo pripojiť k projektoru alebo LCD monitoru cez HDMI port.



Obrázek 3.1: Ukážka aplikácie Remote for PowerPoint[1]

Porovnanie vyvíjanej a existujúcich prezentácií

Aplikácia sa bude snažiť konkurovať komerčným aplikáciám s touto tématikou. Bude obsahovať podmnožinu funkcií z niektorých výššie uvedených aplikácií. Bude umožňovať sledovanie doby na prezentáciu, ktorá uplynula podobne ako aplikácia Remote for PowerPoint, avšak nebude sa môcť pripojiť k PC a ovládať prepínanie slidov. Z aplikácie Presentation Timer bude tiež zdielať niektoré funkcie. Rovnako ako táto aplikácia bude upozorňovať užívateľa na blížiaci sa koniec. Zatiaľ čo však táto aplikácia upozorňuje vizuálne, vyvíjaná aplikácia sa bude orientovať na zvukové a vibračné upozornenie. Vyvíjaná aplikácia bude viacej jednoúčelová ako bežne dostupné komerčné aplikácie, ktoré sa snažia prilákať užívateľov mnohými funkciami. Výhodou oproti ostatným aplikáciám bude však nižšia pamäťová náročnosť. Pokiaľ bude užívateľ vyžadovať iba funkcionalitu pokrytú v špecifikácii požiadavkov, bude preň ho táto aplikácia lepšou voľbou. Aplikácia bude umožňovať navyše oproti ostatným aplikáciám lepšie preusporiadavanie prezentácií, ktoré v existujúcich aplikáciách podobného typu chýbajú. Umožňuje usporiadať bloky prezentácií v ľubovoľnom poradí a časy začiatkov prezentácií sa prepočítajú tak, aby následovali hneď za sebou. V aplikácii bude ako blok braná aj prestávka. Teda organizátori si budú môcť presunúť na neskôr nie len danú prezentáciu, ale aj posunúť prestávku. V móde pre prezentujúcich bude navyše oproti ostatným aplikáciám zobrazovať priebeh akutálneho snímku. Pri manuálnom preskakovaní snímkov (v prípade nedodržania zobrazeného snímku) aplikácia automaticky prepočíta priemerný čas na každý ďalší snímok tak, aby sa stihla prezentácia odprezentovať včas.

Kapitola 4

Návrh a implementácia aplikácie

Aplikácia je určená pre mobilné zariadenia s operačným systémom Android. To určuje špecifiká návrhu aplikácie.

Vytváranie aplikácie bude pozostávať z návrhu obrazoviek a ich vzájomných vzťahov, z vytvorenia návrhových XML súborov pre návrh grafického užívateľského rozhrania, návrhu databázových tabuliek na uloženie perzistentných dát a nakoniec implementácie príslušných aplikačných blokov. Návrh sa bude zaoberať aplikačným blokom *aktivita*, ktorý reprezentuje vždy jednu obrazovku aplikácie a akciami, na ktoré bude aktivita reagovať. Typickými akciami užívateľov sú: kliknutie na určený prvok, dlhé podržanie, posúvanie zoznamu a presunutie prvku systémom drag and drop. Každá akcia vyvolá príslušnú udalosť, v ktorej sa reaguje na užívateľovu požiadavku. Pri kliknutí na softwarové alebo hardwarové tlačítko menu sa napríklad vyvolá vysúvacie menu. Udalosť mení stav aktuálnej aktivity alebo definuje zámer, ktorý obslúži iná aktivita. Pri definovaní zámeru sa aplikácia prepne do ďalšieho okna.

4.1 Návrh obrazoviek

V tejto kapitole bude popísaný konceptuálny návrh obrazoviek aplikácie pre oba módy. Návrh sa neopiera o výber konkrétnych grafických prvkov a ich vzhľad v obrazovkách, ale o obsah a funkcie jednotlivých obrazoviek a ich vzťahy.

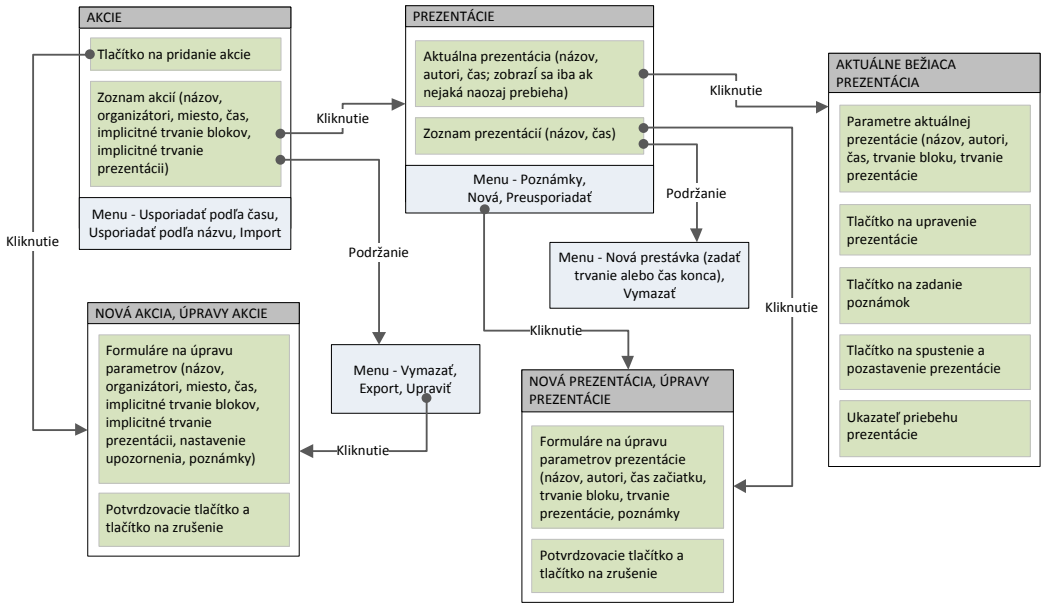
4.1.1 Mód pre organizátora prezentacií

V móde pre prezentujúceho sa nachádzajú 2 základné obrazovky pre organizáciu prezentácií. Jednou je obrazovka akcií, v ktorej je zoznam všetkých naimportovaných alebo vytvorených akcií a druhou je obrazovka konkrétnej akcie a jej naplánovaných prezentácií. Konkrétny popis týchto okien je uvedený v nasledujúcich podkapitolách.

Základný princíp ovládania aplikácie je nasledovný. Pri kliknutí na konkrétny prvok sa zobrazí jeho obsah, dlhé podržanie prsta nad objektom vyvoláva kontextové menu s možnými akciami. Pre každé okno je k dispozícii aj výsuvné menu s akciami, ktoré sa neviažu ku konkrétnemu prvku na obrazovke.

V grafe 4.1 sú schématicky zobrazené obsahy jednotlivých obrazoviek ako aj ich vzájomné vzťahy (akou akciou sa dostanem do inej aktivity). Šípky znázorňujú, akou akciou sa v akej obrazovke dostanem na ďalšiu obrazovku. V grafe sú znázornené dva typy akcií – kliknutie a podržanie. Obrazovky sú štrukturované. Šú tu odlížené bežné prvky, ktoré sa zobrazia na obrazovke – bledo-zeleným pozadím, a menu voľby – bledo-modrým pozadím.

Menu celej obrazovky sa nachádza (vyvolané po stlačení menu tlačítka) vždy v spodnej časti obrazovky, ako býva aj zvyčajne vykreslované. Kontextové menu je zakreslené samostatne mimo obrazovky. Prepnutia do nadradých obrazoviek nie je v grafe naznačené kvôli prehľadnosti. Na tento účel slúži hardwarové alebo softwarové tlačítko „späť“ ako je to uvedené v kapitole 2.1.



Obrázek 4.1: Návrh obrazoviek pre mód organizátorov

Okno akcií

V obrazovke sa zobrazí zoznam naplánovaných akcií, ktoré boli vytvorené priamo v danej aplikácii alebo importované nahraním príslušného CSV súboru a tlačítko na vytvorenie akcie priamo v aplikácii.

Pri vytvorení alebo upravení akcie sa vyvolá okno na zadanie parametrov akcie. Parametre akcie, ktoré sa tu dajú nastaviť sú „názov“, „organizátori“, „miesto“, „začiatok“, „implicitné trvanie blokov“, „implicitné trvanie prezentácií“, „čas upozornenia“ a „poznámky“. Trvanie blokov prezentácií je celkový čas potrebný na samotnú prezentáciu ako aj na diskusiu o práci, otázky a ohodnotenie práce. Čas upozornenia určuje ako dlho pred koncom prezentácie majú byť organizátori upozornení na blížiaci sa koniec prezentácie. Väčšina atribútov akcie je povinná. Jediným nepovinným parametrom sú „poznámky“. Povinný atribút „čas upozornenia“ sa vypočíta implicitne po nastavení času na prezentáciu. Implicitne je táto hodnota nastavená na poslednú štvrtinu prezentácie a dá sa modifikovať na čas v intervale <0; „čas prezentácie“ / 2> pre koncom prezentácie. Teda upozornenie môže nastať vždy iba v druhej polovici prezentácie čo odpovedá významu tohto parametru - upozorniť na blížiaci sa koniec prezentácie kvôli dodržaniu časového harmonogramu.

Akcia má v zozname akcií vypísané všetky povinné atribúty. Po stlačení tlačítka menu sa bude dať určiť usporiadanie zoznamu podľa času alebo názvu akcie a nastaviť akým spôsobom sa bude upozorňovať na ukončenie prezentácie. V zozname sa podržaním prstu

nad vybranou akciou vyvolá ďalšie menu, kde je možné upraviť alebo vymazať vytvorenú akciu.

Okno prezentácií

Po kliknutí na konkrétnu akciu sa zobrazí zoznam prezentácií v poradí v akom následujú za sebou v čase a v akom boli naplánované. Medzi prezentáciami môžu byť vložené aj prestávky. V menu sa nachádzajú voľby pre pridanie novej prezentácie, vloženie poznámok k akcii a preusporiadanie prezentácií. Pri podržaní prsta nad vybraným prvkom sa dá vytvoriť pred vybraným blokom prestávka alebo vymazať určený blok (prezentáciu alebo prestávku). Pre vloženie prestávky existujú 2 spôsoby vloženia: „prestávka s trvaním“ a „prestávka do času“. Pri prvej možnosti sa vytvorí prestávka so zadaným trvaním, druhým spôsobom sa určí čas dokedy má prestávka trvať a samotné trvanie prestávky sa vypočíta automaticky zo začiatku prestávky. Toto okno je interaktívne a zobrazuje aktuálne prebiehajúcu prezentáciu ak existuje. Inak okno zobrazuje iba zoznam prezentácií pre ušetrenie miesta na obrazovke.

Po kliknutí na prezentáciu v zozname prezentácií (nie na aktuálne prebiehajúcu prezentáciu) sa zobrazí editačné okno prezentácie. V tomto okne sa nastavujú parametre prezentácie, ktoré sú: „názov“, „autori“, „začiatok“, „trvanie blokov“, „trvanie prezentácií“ a „poznámky“. Jediným voliteľným parametrom sú poznámky. Pri vytvorení novej akcie je nutné nastaviť všetky povinné parametre, avšak atribúty „trvanie blokov“ a „trvanie prezentácií“ sa predvypĺňajú z rovnomenného atribútu v akcii, v ktorej sa daná prezentácia nachádza a majú aj rovnaký význam. V akcii sa iba definujú implicitné hodnoty týchto parametrov.

Užívateľ môže prostredníctvom popup menu preusporiadať a presunúť danú prezentáciu alebo prestávku medzi iné bloky a tak pružne reagovať na zmeny v harmonograme. Trvanie jednotlivých prestávok a prezentácií ostáva zachované ako bolo nastavené.

Okno aktuálne bežiacej prezentácie

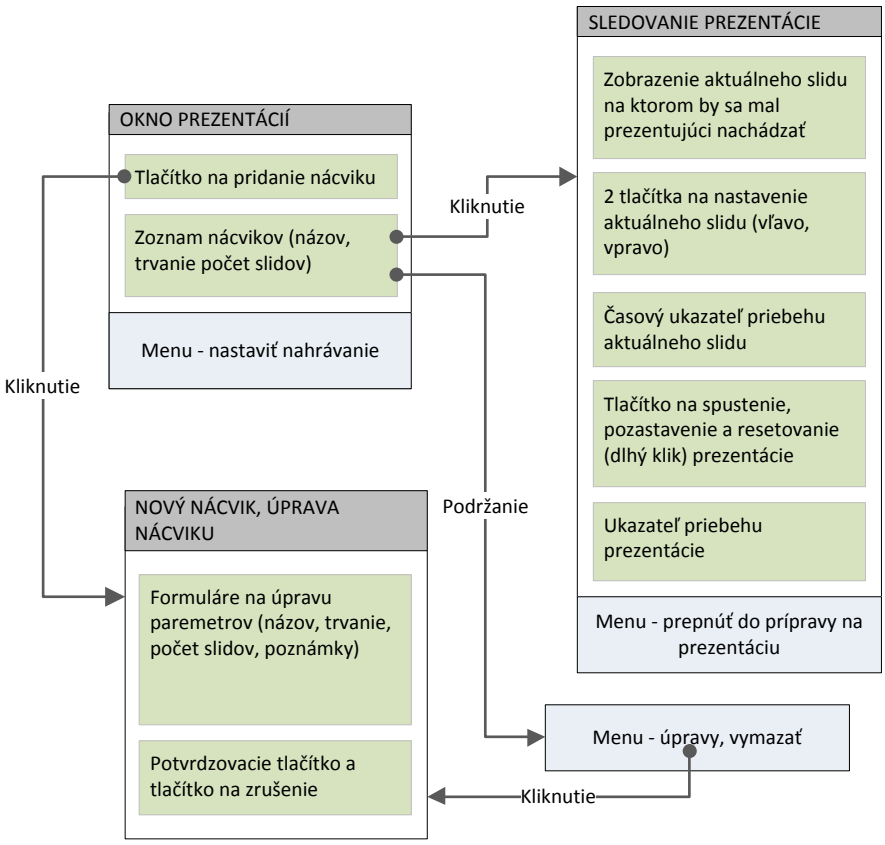
Okno aktuálne bežiacej prezentácie zobrazuje informácie o aktuálnej prezentácii. Zobrazuje autorov prezentácie, začiatok prezentácie, časový priebeh a poznámky. Umožňuje editovať a pridávať nové poznámky, pozastaviť meranie času prezentácie (napr. v prípade technických problémov) a zadať hodnotenie prezentácie. Upozorňuje organizátorov v stanovenom čase na blížiaci sa koniec prezentácie.

4.1.2 Mód pre prezentujúcich

Mód pre prezentujúcich sa skladá z dvoch hlavných okien. Okna, ktoré zobrazuje zoznam všetkých nácvikov a okna, ktoré upozorňuje prezentujúceho počas prezentácie. V obrázku 4.2 je graficky znázornený obsah týchto okien a ich vzťahy, a v nasledujúcich podkapitolách sú tieto okná podrobne popísané.

Okno nácvikov

Toto okno obsahuje zoznam vytvorených a uložených nácvikov. Každý nácvik bude obsahovať atribúty „názov“, „trvanie“, „počet snímkov“ a „poznámky“. Okrem poznámok budú všetky atribúty povinné a prehľadne zobrazené v tomto zozname. Tlačítkom na pridanie sa budú môcť vkladať nové nácviky do zoznamu a podržaním prsta nad vybraným nácvikom



Obrázek 4.2: Návrh obrazoviek módu pre prezentujúcich

v zozname budú umožnené úpravy a vymazanie daných nácvikov. Po kliknutí na príslušný nácvik sa zobrazí okno pre sledovanie času prezentácie.

Okno pre sledovanie času prezentácie

V tomto okne sa bude nachádzať tlačítko na spustenie merania času prezentácie. Opätovným stlačením bude priebeh prezentácie pozastavený a dlhým podržaním resetovaný a vrátený na začiatok. V aplikácii sa bude prehľadne zobrazovať časový priebeh ako celej prezentácie tak aj optimálneho času určeného na konkrétny snímok. V okne bude zobrazené číslo snímku, na ktorom by sa mal prezentujúci v ideálnom prípade nachádzať. Ak sa prezentujúci kvôli časovému sklzu nenachádza na danom snímku, môže si pre presnejšie navigovanie aktuálne číslo snímku posunúť (umožnené obomi smermi - aj keď má naopak viacej času) a aplikácia prepočíta podľa zostávajúceho času a počtu snímkov priemernú dobu strávenú na jednom snímku.

Okno pre prípravu na prezentáciu

Okno obsahuje tlačítko pre spustenie nahrávania a merania času prezentácie. Ďalším tlačítkom dáva užívateľ najavo, že v danom čase prechádza na nasledujúci snímok. Pri prepnutí

na ďalší snímok je zaznamenaný v hornej časti obrazovky čas strávený na predchádzajúcom snímku. V zozname sa nachádzajú všetky prejdené snímky. Po ukončení nahrávania (prejdení na posledný snímok a stlačením tlačítka pre posun snímku) sa zobrazí celkový čas prezentácie a umožní prepočet časov potrebných na jednotlivé snímky tak, aby bol celkový čas prezentácie rovný zadanému optimálnemu času.

4.2 Návrh rozloženia grafických prvkov

Jedným z kľúčových bodov zadania práce je požiadavok na vysokú ergonómiu užívateľského rozhrania. Z tohto požiadavku sa snaží návrh aplikácie vychádzať. Aplikácia nepoužíva exotické a neznáme prvky užívateľského rozhrania a používa štandardné princípy ovládania aplikácie aké sú využité vo väčšine aplikácií, aby si používateľ rýchlo zvykol na systém ovládania a mohol začať využívať program. Aplikácia by mala byť intuitívna a funkcia jednotlivých prvkov na obrazovke jednoznačná. Toto by malo pomôcť užívateľovi zjednodušiť a urýchliť prácu s aplikáciou.

Návrh grafického užívateľského rozhrania a rozloženie ovládacích prvkov v obrazovkách bol realizovaný najskôr prostredníctvom nákresu na papier. Takto nakreslené obrazovky boli potom prevedené do odpovedajúcich návrhových XML súborov umiestnených v pevne danom adresári `/res/layout`. Android pomocou týchto súborov umožňuje jednoduchšie a transparentnejšie vytvorenie stromovej štruktúry prvkov `View`[9], ako keby sme ich mali písať v Java kóde. Oddeľuje sa tým aj aplikačná logika od vzhľadu a preto je aplikácia jednoduchšie upravovateľná a umožňuje oddeliť grafickú prácu a kódovanie logiky. Výstupom tejto etapy vytvárania projektu sú XML návrhové súbory, ktorých obrazovky sú znázornené v prílohe A. Pôvodný náčrt bol prispôsobený možnostiam operačného systému a veľkosti obrazovky. Bol nastavnený vzhľad prvkov, ich hrúbka, farba a font písma. Pri výbere farebných kombinácií bol braný ohľad na zachovanie kontrastov medzi farbou pozadia a farbou popredia (farba písma) tak, aby bol text dobre čiteľný. Aplikácia používa pre vykreslovanie obmedzené množstvo farieb, aby zbytočne nemiatli užívateľa a bolo hneď od začiatku jasné aký význam a váhu majú prvky s danou farbou (napr. červená farba pri písme označuje dôležitý text, šedá zasa doplnkový).

Obrazovky sú navrhnuté iba pre vertikálne natočenie displeja a grafické prvky sú používané s relatívnymi mierami, teda mali by sa zobraziť rovnako aj na displejoch s rozdielnou hustotou pixelov.

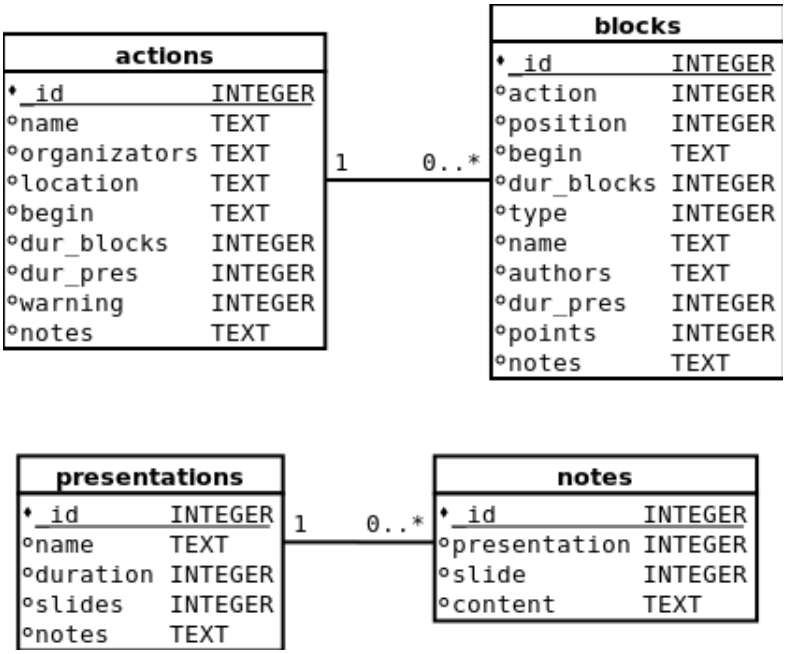
Aplikácia využíva štandardné grafické prvky systému Android pre tlačítko (`Button`), popisok (`TextView`), obrázok (`ImageView`) alebo editačné pole (`EditText`). Ďalej využíva na vykreslenie zoznamu `ListView` prvok spolu s aktivitou zdedenou z triedy `ListActivity`. Na určenie rozmiestnenia jednotlivých prvkov na obrazovke sú využívané návrhové kontajnery `LinearLayout` a `RelativeLayout`, ktoré určujú usoriadanie vložených prvkov.

Ďalej sú v hojnej miere kvôli zameraniu aplikácie využité štandardné ukazatele priebehu. V Androide na to slúžia elementy `ProgressBar` a `SeekBar`. Jediný rozdiel mezdi nimi je ten, že posuvný panel(`SeekBar`) obsahuje aj grafickú úchytku na nastavenie pokroku v paneli.

Aplikácia používa obrázky (`ImageView`) umiestnené v zložke `/res/drawable` na tlačítka spustenia a pozastavenie prezentácie.

4.3 Práca s databázou

Aplikácia bude pracovať s databázou SQLite, ktorá je vstavaná v systéme Android. V tejto kapitole je popísaný návrh databázových tabuliek tak, aby umožnil ukladať celý dátový model aplikácie. Na nasledujúcom obrázku sú tieto tabuľky graficky znázornené.



Obrázek 4.3: Návrh databázových tabuliek

Tabuľka akcií

Tabuľka `actions` uchováva všetky akcie vytvorené alebo naimportované v aplikácii, v móde pre organizátorov prezentácií. Každá akcia má svoje meno, zoznam organizátorov (ich mená oddelené čiarkami), miesto konania, čas začiatku, implicitné trvanie blokov prezentácií, implicitné trvanie prednášok a poznámky. Tomu odpovedajú atribúty `name`, `organizators`, `location`, `begin`, `dur_blocks`, `dur_pres`, `notes` (v tomto poradí) v databázovej tabuľke. Trvanie blokov je čas potrebný na prezentáciu s jej dodatočnou réžiou (čas na otázky, diskusiu, hodnotenie). Ďalej je v akcii možné nastaviť kedy bude užívateľ upozornený na koniec prezentácie (atribút `warning`).

Primárnym kľúčom je atribút `_id` čo je číselný identifikátor danej akcie. Atribúty `name`, `organizators location`, `begin` a `notes` sú v SQLite databáze definované pod typom TEXT a sú to teda textové reťazce. Atribút `begin` reprezentuje začiatok danej akcie vo formáte YYYY-MM-DD HH:MM, kde YYYY je rok, MM je mesiac, DD je deň v mesiaci a HH:MM je čas v 24-hodinovom formáte. Ďalšie stĺpce `dur_blocks` a `bur_pres` sú celočíselného typu (v databáze ako `INTEGER`) a udávajú implicitnú dĺžku blokov a dĺžku prezentácií v minutách. Stĺpec `warning` je číslo od 0–100 a definuje kedy má byť organizátor upozornený na blížiaci sa koniec prezentácie vzhľadom na dĺžku prezentácie. Číslo 0 znamená upozornenie na úplnom konci prezentácie a číslo 100 značí upozornenie v polovici času potrebného na prezentáciu.

Tabuľka blokov

Tabuľka `blocks` obsahuje zoznam prezentácií a prestávok v konkrétnej akcii. Každý blok má jednoznačný identifikátor v rámci celej akcie, nadradenú akciu, v ktorej sa nachádza, pozíciu v zozname prezentácií, čas začiatku, trvanie a typ, či sa jedná o prezentáciu alebo prestávku. Príslušné atribúty v databázovej tabuľke sú `_id`, `action`, `position`, `begin`, `dur_blocks` a `type`. Každá prezentácia obsahuje aj meno, zoznam autorov, trvanie prezentácie a poznámky k prezentácii. Sú to atribúty `name`, `authors`, `dur_pres` a `notes`. V prípade prestávky sa tieto stĺpce neberú do úvahy, a tak ich nie je nutné ukladať.

Primárnym kľúčom je identifikátor `_id`. Atribút `action` nie je cudzí kľúč, pretože zabudovaná SQLite databáza v Androide toto integritné obmedzenie priamo neposkytuje. Toto obmedzenie je ošetrené na úrovni aplikačnej logiky. Do stĺpca sa teda nevkladá to čo nemá, aby bola aplikácia stabilná. Atribúty `begin` a `dur_blocks` sa používajú rovnakým spôsobom ako v tabuľke `actions`. Atribút `type` je číslo 0 v prípade prezentácie a 1 pokiaľ sa jedná o prestávku na odlíšenie týchto typov blokov.

Tabuľka prezentácií

Každá prezentácia obsahuje atribúty názov, trvanie, počet snímkov a poznámky. Príslušné stĺpce v tabuľke `presentations` sú `name`, `duration`, `slides` a `notes` (v rovnakom poradí). Atribút `name` je typu `TEXT`, `duration` typu `INTEGER` a obsahuje trvanie prezentácie v minutách. Ďalšie atribút `slides` je celočíselný a `notes` je deklarovaný s typom `TEXT`. Pri vytvorení nového riadku sú povinné všetky stĺpce okrem stĺpca s poznámkami. Primárnym kľúčom v tejto tabulke je atribút `_id`, ktorý je deklarovaný s kľúčovým slovom `AUTOINCREMENT` pre zaistenie automatickej správy identifikátorov (pri vytvorení riadku sa nájde najmenší nepoužitý identifikátor).

Implementácia databázovej triedy

Na pripojenie k databáze SQLite v systéme Android bola vytvorená trieda `Data`, ktorá reprezentuje aplikačné rozhranie k databáze. trieda `Data` je trieda, ktorá dedí z triedy, `SQLiteOpenHelper` a ktorá zabezpečuje vytvorenie a aktualizáciu databázových tabuliek a prípadné vloženie implicitných stĺpcov do nich. Na získanie instancie triedy `SQLiteDatabase` je treba zavolať metódu `getWritableDatabase()` nad triedou `Data`.

Pri výbere viacerých riadkov z databázy sa používa trieda `Cursor`, ktorá slúži na prechádzanie týchto riadkov. Metódou `moveToFirst()` sa nastaví kurzor na prvý vyhovujúci riadok vrátený SQL dotazom, metóda `moveToNext()` posunie kurzor na ďalší riadok. Kurzor treba po použití zavrieť metódou `close()` na uvoľnenie prostriedkov, alebo označit ako automanažovateľný a vtedy sa o jeho uvoľnenie postará aktivita, v ktorej sa používa. [12, 13]

Testovanie obsahu databázy

V systéme Android bohužiaľ neexistuje jednoduchý spôsob ako zobraziť obsah SQLite databázy aplikácie v mobilnom zariadení. Tento problém som riešil následovne. Prvým krokom bola emulácia mobilného zariadenia prostredím Eclipse s prídavním pluginom ADT Plugin od spoločnosti Google. Vo okne pre virtuálne mobilné zariadenie sa dá zobraziť obsah koreňového adresára a vyhľadať príslušný súbor s uloženou databázou danej aplikácie. Tento súbor sa nachádza v zložke `/data/data/<koreňový balíček aplikácie>/databases/` a

má názov `data.db`. Následne som otvoril tento súbor v aplikácii SQLite database browser, ktorá zobrazuje obsah databázy – vložené tabuľky a ich naplnené riadky.

Týmto postupom som sledoval obsah databázy po vkladaní a mazaní položiek, aby som overil, že sa po nich databáza aktualizuje správne.

4.4 Návrh aktivity

Základným stavebným blokom aplikácie v Androide sú aktivity, ktoré obsahujú prevažnú väčšinu logiky aplikácie. Umožňujú interakciu s používateľom pomocou grafických návrhov uložených v XML súboroch. Reagujú na udalosti vyvolané manipuláciou s dotykovou obrazovkou a zobrazujú na ňu svoj stav. Pri implementácii projektu je teda potrebné navrhnúť základnú štruktúru aktivity.

Aktivita je trieda, ktorá dedí od Android triedy `Activity` alebo jej podtried. Veľmi často využívaná podtrieda je `ListActivity`, ktorá umožňuje jednoduchšie písanie aktivít so zoznamami `ListView`. Na fungovanie triedy musí byť použitý návrhový XML súbor so zoznamom, ktorý má identifikátor prostriedku `android:list`. Pri implementovaní aktivít som prepisoval callback metódy nadradených tried, ktoré sú popísané v kapitole 2.1.

Metóda onCreate()

V metóde `onCreate()`, ktorá je volaná pri vytvorení aktivity sa nastavuje rozloženie grafických prvkov na obrazovke pomocou metódy `setContentView()`. Tejto metóde je predaný identifikátor návrhového XML súboru. Metóda bola takisto využívaná často na inicializáciu prístupu k databáze, ktorý sa používal v ďalších metódach aktivity na získavanie a zapisovanie dát do databázy. V metóde sa ďalej inicializujú dátové členy aktivity, ktoré uchovávajú prvky grafického užíteľského rozhrania danej obrazovky. Prostredníctvom týchto premenných môže aplikácia príjmať užívateľom zadané informácie a zobrazovať výsledky. Slúžia na komunikáciu s grafickým užívateľským rozhraním. Pri týchto členoch je tu možné nastaviť *poslucháč* (listener), teda metódy, ktoré budú zavolané pri špecifických udalostiach vykonaných nad príslušnými grafickými objektami. Najčastejšie používaná je metóda `addOnClickListener()`, ktorá umožňuje pridať reakciu na stlačenie prvku. Podobnou metódou je `addOnClickListener()` pri dlhom podržaní elementu na obrazovke. Novšie verzie Androidu umožňujú poslucháč pri kliknutí nastaviť aj priamo v návrhovom XML súbore.

V aktivitách so zoznamom `ListView` je potrebné nastaviť adaptér pre tento zoznam. Adaptér je trieda, ktorá poskytuje zoznamu ako dáta, tak aj spôsob ich vykreslenia. Na toto je využitá spomínaná trieda `ListActivity` a metóda `setListAdapter()`. Ako adaptér je využitý `SimpleCursorAdapter`, ktorý získava dáta z databázových tabuliek. Pri vytváraní tohto adaptéru sa zadá zoznam stĺpcov tabuľky databázy a k nemu príslušiaci zoznam identifikátorov príslušných grafických prvkov, do ktorých majú byť jednotlivé stĺpce vykreslené.

Všetky ďalšie inicializácie premenných sa nachádzajú buď v tejto triede, alebo triedach `onStart()` a `onResume()`, ktoré sa volajú po vytvorení aktivity.

Ostatné metódy životného cyklu aktivity

Metódy `onStart()` a `onResume()` sú využívané na aktualizáciu grafického užívateľského rozhrania po zmene dátového modelu. Pri návrate z dialógového okna, ktorý mení obsah databázy sa zavolá v nadradenej aktivite jedna z týchto metód. V metóde je znovu načítaný databázový kurzor do adaptéru zoznamu, aby sa zoznam obnovil a zobrazoval aktuálne

hodnoty z databázy. Metóda `onResume()` je využívaná v situáciách, kde je vyžadovaná častejšia aktualizácia (aktivita nemusí byť stopnutá, iba už nie je na popredí), aj za cenu nižšej výkonnosti.

V prípade metód `onStop()` a `onPause()` sa jedná o opačný prípad. Užívateľ opúšťa danú aktivitu, a tak je nutné namiesto získania nových dát, zadané dáta uložiť. Tento princíp je využitý napríklad pri ukladaní zadaných poznámok po stlačení tlačítka „späť“ v okne *Pomoci pri prezentácii.*

Ďalšie callback metódy

Aktivita používa aj callback metódy, ktoré nesúvisia s jej životným cyklom. Sú to metódy, ktoré sa automaticky volajú napríklad po stlačení menu tlačítka, kliknutí na prvok zoznamu alebo dlhom podržaní prvku na vyvolanie kontextového popup menu.

Veľmi často využívané je kontextové menu. Je znázornené na obrázku A.5. Toto menu sa vyvoláva po podržaní prstu nad objektom dlhšiu dobu. V tomto menu sa zobrazujú možné akcie s vybraným prvkom. Viaže sa k zoznamu `ListView` alebo podobným kontajnerom. Najskôr sa zaregistruje zoznam `ListView` pomocou metódy `registerForContextMenu()` pri spustení aktivity. Na vykreslenie kontextového menu sa prepíše funkcia `onCreateContextMenu()`, do ktorej sa pridávajú voľby, ktoré majú byť k danému prvku zobrazené. Použije sa funkcia `add()` kontextového menu a predá sa jej identifikátor voľby a popisok. Identifikátor voľby je využitý vo funkcii `onContextItemSelected()`, ktorá sa zavolá po výbere príslušnej voľby v kontextovom menu. Na základe identifikátoru sa vykoná príslušná akcia, ktorú užívateľ požaduje.

Ďalšou využívanou callback metódou je metóda `onListItemClick()`, ktorá sa vyvolá po stlačení prvku zoznamu. V prípade `ListActivity` je táto metóda implicitne zaregistrovaná na implicitný ListView s identifikátorom prostriedku `android:id` a stačí ju teda iba implementovať.

Tieto callback metódy sú zavolané s parametrami špecifikujúcimi, nad ktorým konkrétnym prvkom bola prevedená akcia, prípadne aký bol jeho obsah. Môžeme zistiť napríklad číslo riadku databázovej tabuľky, ktorý daný prvok zobrazuje alebo pozíciu na obrazovke.

Metóda `onCreateOptionsMenu()` je vyvolaná po stlační menu tlačítka. V tejto metóde sa podobne ako v prípade kontextového menu pridávajú nové voľby do menu funkciou `add()`. Metóda `onOptionsItemSelected()` je zavolaná po stlačení príslušnej akcie v menu s parametrom typu `MenuItem`, na základe ktorého môžeme vyvolaním metódy `getItemID()` rozlíšiť o akú voľbu sa jednalo a vykonať príslušnú akciu.

Komunikácia medzi aktivitami

Aktivity medzi sebou komunikujú prostredníctvom zámerov a triedy `Intent`. Na spustenie novej aktivity sa využívajú 2 funkcie: `startActivity()` a `startActivityForResult()`. Prvá funkcia spustí rovnocennú aktivitu, ktorá sa do pôvodnej aktivity už nemusí vôbec dostať, druhá zavolá aktivitu ako dialóvé okno. Metóda `startActivityForResult()` očakáva od spustenej aktivity výsledok, ktorý sa má využiť v pôvodnej aktivite. Teda jedná sa o podradenú aktivitu, ktorá existuje iba kvôli získaniu dát pre pôvodnú aktivitu. V projekte sa vyžívajú obe metódy podľa žiadaného výsledku.

Ak podradená aktivita ukončí svoju činnosť, v pôvodnej aktivite sa zavolá callback metóda `onActivityResult()` a je jej predaný identifikátor ukončenej aktivity. Tento identifikátor sa uvádza ako druhý parameter funkcie `startActivityForResult()`. V metóde `onAcitivityResult()` sa vyvolajú príslušné akcie so získanými dátami.

Pri volaní aktivity sa do aktivity môžu poslať dáta, podľa ktorých môže spustená aktivita prispôsobiť svoje chovanie. Tento princíp je veľmi často využívaný. Môžeme tak spojiť dve aktivity do jednej v niektorých prípadoch. V projekte sa to využíva pri vytvorení a editácii novej akcie či prezentácie. Obrazovky na vytvorenie akcie a editáciu akcie sa líšia iba v tom, že v prípade editácie sú hodnoty predvyplňené zo starých hodnôt. Logika aktivity je odlišná zase iba v jednom databázovom dotaze. V prípade vložení sa použije metóda `insert()`, pri úprave metóda `update()`. Kvôli týmti odlišnostiam nie je nutné vytvárať dve odlišné aktivity, kedže väčšina kôdu by bola rovnaká. Preto je vytvorená jedna aktivita spustená s parametrom `EDIT_PARAM` a prispôsobuje svoje chovanie na základe jeho hodnoty. Na spustenie aktivity s parametrom sa zavolá funkcia `putExtra()` nad objektom typu `Intent` ešte pred spustením funkciou `startActivity()`.

Na vloženie návratovej hodnoty v podradenej aktivite slúži funkcia `setResult()` a na samotný návrat funkcia `finish()`. Nie je teda nutné uvádzať meno nadradenej aktivity, ktorá sa má spustiť. Systém si túto informáciu zapamätá automaticky za nás pri vytvorení aktivity funkciou `startActivityForResult()`. Po obnovení pôvodnej nadradenej aktivity a presunutí do popredia sa vyvolá jej funkcia `onActivityResult()`, do ktorej je vložený parameter `requestCode` s identifikátorom predtým spustenej podaktivity a `resultCode` s návratovou hodnotou. Funkcia môže definovať aj viacej návratových hodnôt a s rôznymi typmi. Stačí vytvoriť získať zámer aktivity funkciou `getIntent()`, vložiť doň ho podobne ako pri volaní aktivity požadovanú hodnotu metódou `putExtra()`, ktorá má prvý parameter typu `String` a identifikuje názov parametru a druhý parameter typu `int`, `String`, `double` apod. s vloženou hodnotou.

Na získanie vložených hodnôt či už v podradenej alebo nadradenej aktivite slúžia metódy `get<typ>Extras()` alebo sa dajú nahradiť dvojicou zreťazených metód `getExtras().get<typ>()`. Komunikácia medzi nadradenou a podradenou aktivitou je vždy uložená v zámere podradenej aktivity. V prípade podradenej aktivity tento zámer získame metódou `getIntent()`, v prípade nadradenej aktivity je príslušný intent podradenej aktivity vrátený v parametre callback metódy `onActivityResult()`.

4.5 Implementácia aktivít

V tejto kapitole bude popísaná implementácia hlavných aktivít daných módov. Vychádza z predošlej kapitoly *Návrh aktivity*, kde sa konceptuálne riešilo rozvrhnutie logiky aktivity do jednotlivých callback metód. Táto kapitola rozširuje tento návrh a podrobne popisuje obsah jednotlivých hlavných metód v aktivite.

4.5.1 Mód pre organizátorov

Nasledujúci text popisuje návrh aktivít, využítých pri móde pre organizátorov. Text je členený najskôr podľa obrazoviek aktivít, ďalšie podkapitoly sa podrobnejšie venujú určitým špecifickým problémom v aplikácii.

Hlavné okno aplikácie

Mód pre organizátorov prezentácií sa spúšťa spustením aktivity `PresentationManager`. Táto trieda dedí od `ListActivity` a obsahuje v návrhovom XML súbore obrazovky element `ListView`, ktorý slúži na zobrazovanie zoznamu akcií v danom okne. Pri úprave alebo pridaní akcie je zavolaná aktivita `ActionDialog`, ktorá obsahuje forumulár na nastavovanie

atribútov akcie. Do tejto aktivity sa vkladá v parametre `PARAM_EDIT_MODE` typ spustenia (nová akcia alebo editácia existujúcej akcie). Ďalším parametrom je `PARAM_ACTION`, ktorý sa využíva iba v prípade úpravy aktivity a určuje identifikátor upravovanej akcie. Okno s akciami obsahuje kontextové menu a výsuvné menu. Na ich vytvorenie boli využité princípy z kapitoly 4.4. Do kontextového menu boli pridané metódou `add()` voľby „Upraviť“, „Vymazať“ a „Export“ podľa návrhu obrazoviek v kapitole 4.1. Výsuvné menu obsahuje voľby „Usporiadať podľa času“, „ Usporiadať podľa názvu“ a „Import“. Voľby na usporiadanie zoznamu využívajú na usporadanie prvkov SQL databázový dotaz s klauzulou `ORDER BY`. Importu a exportu sa venuje samostatná podkapitola.

Okno prezentácií konkrétnej akcie

Po kliknutí na prvok zoznamu akcií, je v príslušnom poslucháči vytvorený zámer na vyvolanie aktivity konkrétnej akcie. Táto aktivita je v súbore `Action`. Jediným parametrom tejto aktivity je identifikátor akcie, ktorej zoznam prednášok má byť zobrazený. Aktivita si pri spustení zistí všetky potrebné atribúty danej akcie z dodaného identifikátoru akcie funkciou `getActionParameters()`. Vytvorí adaptér z databázového kurzora, ktorý obsahuje bloky danej akcie. Použitý je adaptér typu `MySimpleCursorAdapter`, ktorý dedí z triedy `SimpleCursorAdapter` a prepisuje metódu `getView()`, ktorá sa používa na vykreslovanie jednotlivých riadkov zoznamu z dodaných dát. V tejto funkcii sa nastavuje pri databázovom riadku s prezentáciou obrázok prezentácie a pri riadku s prestávkou obrázok kávy na grafické odlíšenie týchto typov blokov na obrazovke. Vytvorený adaptér sa použije ako adaptér zoznamu `ListView`. Po kliknutí na voľbu „Nová“ v menu obrazovky alebo po kliknutí na prvok zoznamu je vytvorené dialógové okno pre vytvorenie a úpravu prezentácie s názvom `PresentationDialog`. V menu obrazovky sa nachádza voľba pre zobrazenie poznámok akcie a voľba na vytvorenie dialógového okna `DragNDropListActivity`, ktorý umožňuje preusporiadanie prezentácií. Preusporiadavanie prezentácií je vysvetlené v samostatnej podkapitole kvôli prehľadnosti. Kontextové menu zoznamu obsahuje voľby „Nová prestávka s trvaním...“, „Nová prestávka do času...“ a „Vymazať“ so sémantikou určenou v kapitole 4.1. Po kliknutí na prestávku sa umožní nastavenie trvania prestávky. Aktivita `Action` využíva vlákno na zobrazovanie aktuálnej prezentácie nad zoznamom všetkých prezentácií. Spôsob akým to robí je vysvetlený v podkapitole *Využitie paralelného spracovania* kvôli prehľadnosti.

Riešenie importu a exportu

Na import a export bol využitý CSV formát súboru. Logika importu je uvedená v aktivite `PresentationManager` v metóde `makeImport()`. Táto funkcia je zavolaná po stlačení príslušnej voľby v menu obrazovky. Najskôr sa spustí aktivita `FileDialog` na výber súboru, ktorý sa má importovať. Táto aktivita bola prevzatá z SVN repozitára `code.google.com` spoločnosti Google. Aktivita príjma parameter `SELECTION_MODE` kde je určené či sa má dialóg použíť na otvorenie alebo uloženie súboru. Pomocou parametra `FORMAT_FILTER` je nastavený typ súboru, s ktorým sa bude pracovať. Pri importe je nastavený ešte parameter `START_PATH` na určenie adresára, do ktorého sa vstúpi pri spustení aktivity a parameter `CAN_SELECT_DIR`, kde je uvedená hodnota `false`, aby sa nedal označovať celý adresár. Pri návrate z aktivity je vo funkcii `onActivityResult()` nadradenej aktivity získaná cesta k vybranému súboru vo forme reťazca. Ďalej je vyvolaná metóda `makeImport()`, do ktorej je vložená získaná cesta. V tejto metóde sa otvorí súbor s príslušnou cestou a pomocou metódy `isValidCSVFormat()` sa zistí, či je daný súbor validný a môžeme ho naozaj

naimportovať do aplikácie. V metóde sa porovnáva počet stĺpcov, kontrolujú sa hodnoty na jednotlivých stĺpcoch (či odpovedajú typu), správny formát dátumu a zisťuje sa či sú vložené všetky povinné atribúty v správnom poradí. Po tejto kontrole je zavolaná metóda `createTablesFromCSV()`, kde sa vytvorí nový riadok v tabuľke akcií z prvého riadku súboru (metódou `createActionTable()`) a riadky v tabuľke prezentácií z ostatných riadkov importného súboru (metódou `createPresTable()`). Po aktualizácii databázového modelu je zavolaná metóda `refresh()`, ktorá znovu získa a nastaví databázový kurzor do príslušného adaptéru zoznamu, aby sa obnovil a zobrazoval akutálne dáta.

Preusporiadavanie prezentácií

Na organizovanie a preusporiadavanie prezentácií v akcii slúži v programe aktivita `DragNDropListActivity`. Príjma jeden parameter – identifikátor príslušnej akcie. Jedná sa o zoznamovú aktivitu (`ListActivity`), ktorá nastavuje na zoznam adaptér typu `DragNDropAdapter`. Adaptér `DragNDropAdapter` dedí z bázového adaptéru `BaseAdapter`. Zoznam je v návrhovom XML súbore deklarovaný ako `DragNDropListView`. Nejedná sa o bežný `ListView`, ale je prispôsobený pre účely preusporiadavania. Umožňuje nastaviť poslucháča `DragListener` a `DropListener` a tieto poslucháče sú vyvolané pri akcii drag and drop. Obsahuje metódu `onTouchEvent()`, ktorá obsluhuje všetky možné akcie nad zoznamom. Udalosť je zavolaná s parametrom typu `MotionEvent`, ktorý obsahuje typ akcie a súradnice polohy na obrazovke. Typy akcií sú `ACTION_DOWN`, `ACTION_MOVE`, `ACTION_CANCEL` a `ACTION_UP`. Vo funkcii sa pri udalosti `ACTION_DOWN` uloží pozícia prvku a zavolá sa funkcia `startDrag`, do ktorej sa vloží pozícia prvku. Táto funkcia vytvorí nový grafický objekt, ktorý bude reprezentovať presúvaný prvok. Pri akcii `ACTION_MOVE` sa volá funkcia `drag()` so súradnicami dotyku a táto funkcia premiestňuje objekt vytvorený funkciou `startDrag()`. Akcia `ACTION_UP` zneviditeľný prvok na presúvanie a zavolá metódu `onStopDrag()` poslucháča, ktorý sa registruje metódou `setDragListener()`. V aktivite `DragNDropListActivity` sa po stlačení potvrdzujúceho tlačítka aktualizuje obsah databázy podľa usporiadaného zoznamu `DragNDropListView`.

Ukladanie nastavení

Na ukladanie nastavení slúži trieda `EditPreferences`, ktorá dedí z `PreferenceActivity`. Pracuje s XML súborom s koreňovým elementom `PreferenceScreen`. Podradené elementy majú uvedený popisok, ktorý sa má zobraziť užívateľovi a názov preferencie, ktový sa bude využívať ako identifikátor na získanie príslušného nastavenia. V aktivite preferencií sa tento XML súbor nastavuje funkciou `addPreferencesFromResource()`. Táto aktivita si potom sama ukladá zadané nastavenia pri návrate z aktivity. Tieto nastavenia sú globálne a môžu ich zisťovať všetky aktivity aplikácie. Na tento účel sa používa funkcia `PreferenceManager.getDefaultSharedPreferences()`, ktorá vráti objekt typu `SharedPreferences`. Prostredníctvom tohto objektu môžeme získať konkrétne nastavenie podľa názvu preferencie uvedeného v elemente XML súboru. V prípade elementu `CheckBoxPreference` sa využíva napríklad metóda `getBoolean()`, ktorá príjma názov preferencie a implicitnú hodnotu preferencie, ktorá bude použitá v prípade, že nebol tento atribút užívateľom nastavený.

Využitie paralelného spracovania

Táto podkapitola čerpá z knihy *Java 6 – Výukový kurz*[17] od autora Sharona Zakhoura, z knihy *Android 2*[12] a zo stránky *Android Developers – Processes and Threads*[7]. Ukazuje použitie vlákien v riešenej aplikácii.

V aplikácii sa využívajú vlákna na priebežnú aktualizáciu aktuále prebiehajúcej prezentácie (v zelenom rámiku) podľa času. V Správcovi prezentácií sa vytvára nové vlákno pri vytvorení aktivity `Actions`. Najskôr sa vytvorí nový objekt typu `Thread`, do ktorého je pridaný nový objekt implementujúci rozhranie `Runnable`. Tento objekt musí obsahovať metódu `run()`, v ktorej sa nachádza samotný kód vlákna, ktoré sa bude vykonávať s kódom aktivity paralelne. Na komunikáciu s aktivitou je pri vytvorení aktivity nainicializovaný tzv. *uchytávač* typu `Handler`, ktorý prepisuje metódu `handleMessage()`, ktorá je volaná z vytvoreného vlákna. V kóde vlákna sa každé 2 sekundy vyvoláva táto metóda na aktualizáciu užívateľského rozhrania. Tento princíp je využívaný na zobrazovanie aktuálnej prezentácie v okne prezentácií, kde je takto volaná funkcia `setCurrentPresentation()` nachádzajúca sa v uchytávači. Táto funkcia vždy nastavuje aktuálnu prezentáciu podľa aktuálneho času.

Vlákna sa nepriamo používajú aj v obrazovke prebiehajúcej prezentácie v aktivite `Presentation`, kde je využitý element `Chronometer` v návrhovom súbore. Vloženie tohto elementu do návrhového súboru a spustenie chronometra funkciou `start()` vytvorí automaticky nové vlákno, ktoré každú sekundu vyvoláva zaregistrovanú funkciu, v tomto prípade funkciu `onChronometerTick()`. V aktivite `Presentation` sa v tejto metóde inkrementuje krok `ProgressBaru`, ktorý graficky zobrazuje priebeh prezentácie. `Chronometer` sa spúšťa a zastavuje na základe stlačení tlačítka vedľa `ProgressBaru`.

4.5.2 Mód pre prezentujúcich

Hlavná aktivita módu `Main` je zdedená z triedy `ListActivity`, lebo je určená primárne k zobrazeniu zoznamu prezentácií. Návrhový XML súbor obsahuje element `ListView` s identifikátorom `android:listview`, ktorý je vyžadovaný pre triedu `ListActivity`. V metóde `onCreate()` sa nastavuje vytvorený návrhový súbor s rozvrhnutím prvkov na obrazovke. Kontextové menu pri dlhom podržaní prsta nad prvkom zoznamu obsahuje voľby „Upraviť“ a „Vymazať“. Na obrazovke sa nachádza aj tlačítko na pridávanie nových prezentácií, ktoré má priamo v XML súbore definovanú callback metódu. Po kliknutí na tlačítko je vyvolané dialógové okno `RecordDialog`. Rovnaká aktivita je vyvolaná aj po stlačení voľby „Upraviť“ v kontextovom menu. V prípade editácie sa k aktivite pribalí číslo riadku prezentácie v databázovej tabuľke, ktorá sa bude upravovať a aktivita načíta z databázy implicitné hodnoty. Po kliknutí na konkrétnu prezentáciu sa prejde na aktivitu `Timer`.

Táto aktivita reprezentuje jadro aplikácie. V aktivite sú 4 dátové členy: „aktuálny snímok“, „celkový počet snímkov“, „aktuálny krok“, „celkový počet krokov“. Krokom v aplikácii sa rozumie jedna sekunda prezentácie. Pri vytvorení aktivity sa získa databázový helper, zistí číslo prezentácie, ktorú ideme zobrazovať z parametra, získajú sa ukazatele na prístup k grafickým prvkom obrazovky a nastaví sa návrhový XML súbor. Na tlačítko sa nastaví posluchač na dlhé stlačenie (využíva sa na resetovanie časovača). Trieda využíva v XML súbore element `Chronometer`, ktorý funguje ako stopky. S aktivitou sa komunikuje prostredníctvom tlačítok, všetka logika sa nachádza v príslušných callback metódach a ich pomocných metódach. Táto aktivita má nasledujúce dôležité funkcie.

Funkcia setSlide()

Táto funkcia posunie prezentáciu na konkrétny snímok, pričom krok a teda celkový priebeh prezentácie ostáva zachovaný. Prepočíta teda počet krokov na jeden snímok (z priemeru) a nastaví túto hodnotu ako maximálny krok do `ProgressBar`u. Po uplynutí daného počtu sekúnd bude teda ProgressBar snímku na konci ako je vyžadované. Ďalšou akciou je nastavenie priebehu aktuálneho snímku na krok 0. Pri nastavení snímku sa vložia aktuálne poznámky k snímku v `EditText`e do databázy a načíta sa nový obsah poznámok. Táto funkcia sa volá pri manuálnom nastavení snímku stlačením príslušného tlačítka na obrazovke, alebo aj pri posune o jeden snímok vpred či vzad.

Funkcia onPlayClick()

V tejto metóde sa spustí alebo pozastaví chronometer a krokovanie, ktoré prebieha opätovným volaním callback funkcie `onChronometerTick()`. V tejto funkcii sa posúvajú priebehy ako snímkov tak aj celej prezentácie. Pri spustení chronometra je zabránené užívateľovi editovať poznámky (pri prezentácii už musia byť poznámky pripravené) a naopak pri pozastavení je zase editácia umožnená. Po stlačení tlačítka „späť“ sú poznámky k snímkom uložené, lebo sa vkladá obsah aktuálneho snímku aj pri vyvolaní metódy `onPause()`.

Kapitola 5

Testovanie

V tejto kapitole bude popísané akým spôsobom bola testovaná implementovaná aplikácia. Pri aplikácii sa testovalo, či funguje tak, ako sa od nej očakáva podľa špecifikácie požiadavkov. Testovala sa aj praktická použiteľnosť aplikácie. Tu sa zisťovalo, či má daná aplikácia naozaj využitie v reálnom živote a či môže pomôcť pri správe prezentácií.

5.1 Testovanie funkčnosti

Aplikácia je implementovaná pre zariadenia s minimálnou verziou SDK *7*. To znamená, že sa nebude dať nainštalovať na zariadenia s nižšou verziou operačného systému Android, ako je verzia *2.1*. Aplikácia bola pri implementácii testovaná na mobilnom zariadení *HTC Hero* s firmvérom *VillainROM9.0.0*, ktorý optimalizuje základnú verziu operačného systému Android *2.1*. Telefón má rozlíšenie HVGA (320x480). Program bol vyvíjaný na operačnom systéme *Ubuntu 12.04 LTS* v prostredí *Eclipse 3.7.2* s pridaným pluginom *ADT Plugin* od Androidu. Na testovanie obsahu databázy sa používal program *SQLite database browser*, na zachytávanie snímok obrazovky nástroj v Android SDK *DDMS*. Pri vývoji sa ďalej používala konzolová aplikácia adb dostupná štandardne v Android SDK. Pri vývoji boli využité všetky pokročilé nástroje na ladenie aplikácií v programe Eclipse – vstavaný debugger, výpisy v okne *LogCat*, výpisy na štandardnom výstupe apod. Aplikácia bola odskúšaná kvôli testu kompatibility aj na zariadeniach *HTC Wildfire*, *HTC Evo 3D* a na tablete *Samsung GalaxyTab*.

5.2 Zhodnotenie použiteľnosti

V tejto kapitole budú zhrnuté a analyzované výsledky z pokusného testovania aplikácie deviatimi nezávislými respondentami, . Respondenti používali aplikáciu nainštalovanú na mobilnom zariadení *HTC Hero*, s rozlíšením HVGA a operačným systémom Android 2.1. V tomto teste bolo cieľom pravdivo zistiť praktickú využiteľnosť aplikácie, jej intuitívnosť, prehľadnosť grafického užívateľského rozhrania a zistiť, ako rýchlo sa dokážu užívatelia s aplikáciou zoznámiť. Zoznam úloh a otázok, ktoré testeri riešili je uvedený v prílohe B.

5.2.1 Praktické úlohy

Testeri dostali 4 jednoduché praktické úlohy na 5-10 minút s aplikáciou, aby sa zistili slabé miesta aplikácie, a mohol sa zvážiť ďalší vývoj a prispôsobenie aplikácie. Pri týchto

úlohách užívatelia na nič neodpovedali, ale bola sledovaná ich činnosť. Vyhodnotenie týchto pozorovaní sa nachádza v tejto kapitole.

Prvá úloha bola prípravnou úlohou pre nasledujúce úlohy. Užívatelia sa tu mali iba sami zoznámiť s aplikáciou a pokúsiť sa odhadnúť jej zámer. Väčšina testerov správne odhadla význam aplikácie na základe popiskov „Akcia“, „Prezentácia“, „Organizačný mód“, „Prezentačný mód“ alebo obrázkov. Niektorým to ale trvalo trochu dlhšie. Všetci užívatelia mali skúsenosti s princípom ovládania podobných mobilných aplikácií a tak skúšali zobrazovať výsuvné menu v obrazovkách na zobrazenie volieb a klikať na grafické prvky. Menej často využívali akciu podržanie prsta na vyvolanie kontextového menu.

V druhej úlohe už išlo o konkrétnu úlohu s aplikáciou. Po úvodnom zoznámení s aplikáciou mali užívatelia vytvoriť jednu akciu a pridať do nej 2 prezentácie. Táto úloha nerobila testerom problém. Pri testovaní však veľa respondentov miatla čierna obrazovka prázdnej akcie, v ktorej je nutné pridať akciu vyvolaním výsuvného menu.

Tretia úloha bola pre 4/9 respondentov problémová. Títo testeri mali problém rýchlo násť menu s nastaveniami. Ostatné časti zadania však už zvládali dobre.

Posledná úloha prebiehala bez problémov. Užívatelia si už zvykli na základné princípy ovládania z predošlého módu a tak zadanie zvládli za krátky čas.

5.2.2 Výsledok dotazníku

Zúčastnení respondenti sú študentami Fakulty informačných technológií VUT v Brně, a teda všetci sú vhodní kandidáti pre tento dotazník, keďže majú skúsenosti s obajobami projektov prebiehajúcich na tejto fakulte. V dotazníku boli položené 4 otázky, ktoré zisťovali využiteľnosť aplikácie v reálnom prostredí.

V prvej otázke sa zisťovala využiteľnosť nápovedy v aplikácii po prvom spustení aplikácie. V prípade že by bola aplikácia dostatočne intuitívna, zbytočne by užívateľov pri prvom spustení oberala o čas. Výsledok sa prikláňa k vytvoreniu užívateľskej nápovedy, keďže by užívateľom uľahčila prvotné zoznamovanie sa s programom.

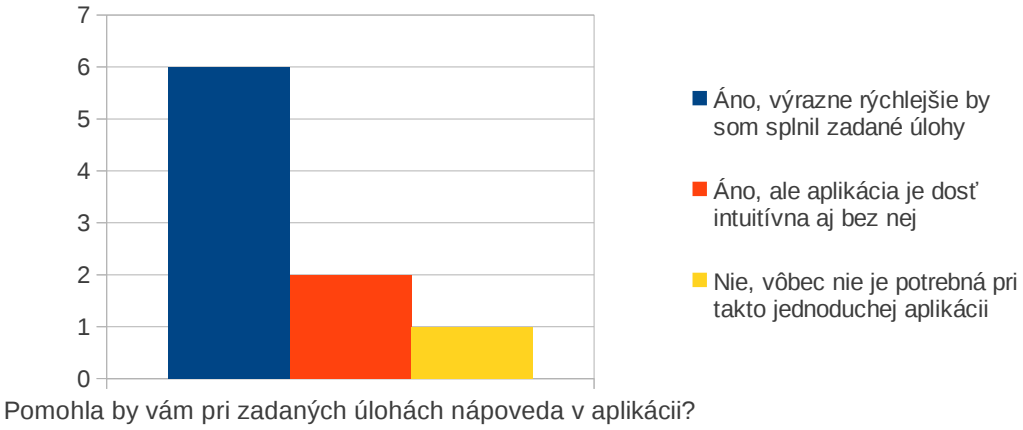
V otázke 2 sa zisťovalo, ktorým problémom pri prezentáciách sa treba v ďalšom vývoji zaoberať najviac, keďže trápi najväčšie percento prezentujúcich. V tejto otázke bolo možné narozdiel od ostatných označiť aj viacej odpovedí. Najčastejším problémom bolo zabudnutie odprezentovania nejakého pripraveného bodu prezentácie. Nedodržanie stanoveného času na prezentáciu bolo málo časté, avšak súvisí s možnosťou A, kde prezentujúci kvôli nedostatku času nestihne pokryť všetky body prezentácie.

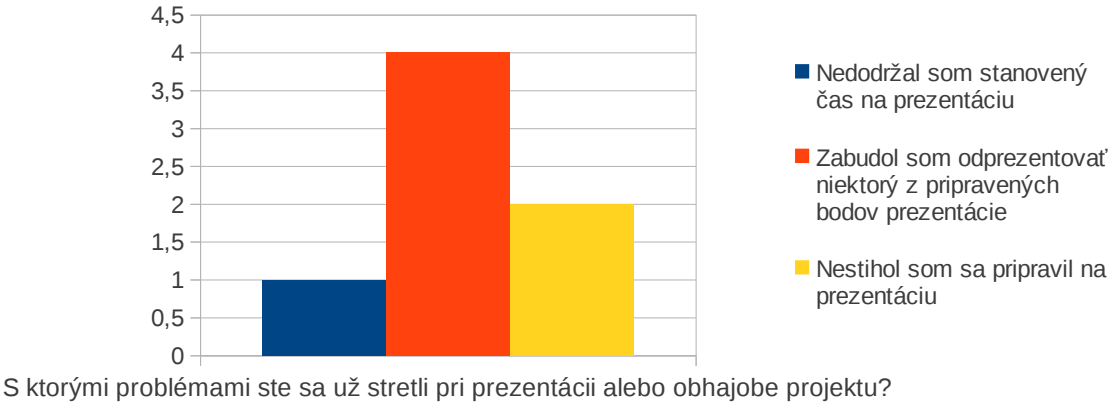
Tretia otázka zisťovala samotnú praktickú použiteľnosť aplikácie a či má pre užívateľov význam. Väčšina respondentov vy túto aplikáciu využila pri prezentácii a organizácii obhajob, v prípade, že by nenašli na Android Markete lepšiu alternatívu. Z tohto výsledku vyplýva, že aplikácia je vyžiteľná v praxi a má význam pri správe prezentácií.

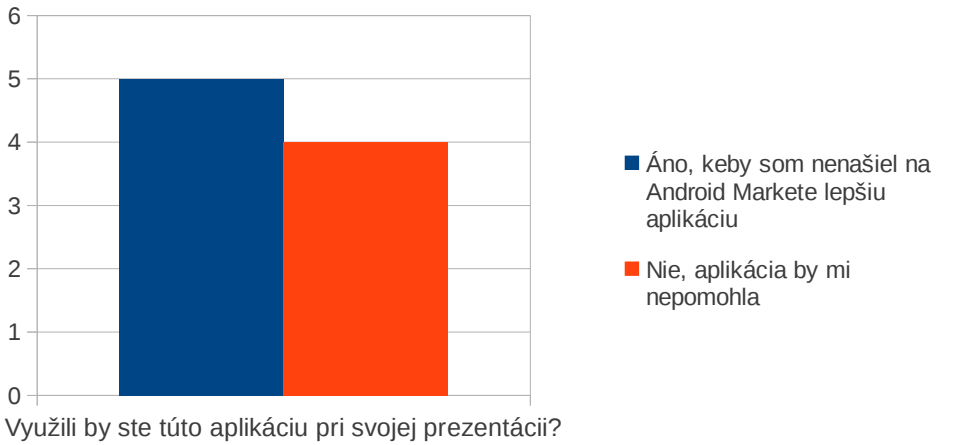
V poslednej otázke respondenti písali svoje návrhy na vylepšenie aplikácie na základe skúsenosti z praktických testov. Užívatelia navrhovali hlavne vylepšenie grafiky aplikácie, ktorá bola pre nich príliž jednoduchá a málo atraktívna. V móde pre organizátora prezentácií navrhovali umožňovať nastavovanie času upozornenia priamo v obrazovke aktuálnej prezentácie a využitie svetelných diód v telefóne na upozornenie. V móde pre prezentujúcich by vylepšili nastavovanie pomerov časov na jednotlivé snímky a prijali by možnosť kreslenia do poznámok, alebo vkladanie obrázkov.

5.2.3 Zhrnutie výsledkov

Z výsledkov testovania môžeme povedať, že aplikácia má pre určitú skupinu užívateľov význam a takisto je aj jednoducho používateľná a intuitívna. Pri praktických úlohách sa zistilo, akú dobu je treba na zoznámenie sa s aplikáciou. Skúsený používateľ zvládne tieto úlohy za približne 2 minúty (okrem prvej, tam je čas na zoznámenie pevne stanovený). Novým užívateľom trvali tieto úlohy dvakrát dlhšie. Po ukončení testovania boli však oboznámený s väčšinou dostupných funkcií aplikácie. Z pozorovania testov a analýzy odpovedí dotazníka boli zistené slabé miesta aplikácie a navrhnuté možné vylepšenia do budúcna.







Obrázek 5.1: Grafy s výsledkami dotazníku B

Kapitola 6

Záver

Cieľom tejto práce bolo zoznámenie sa s platformou Android a implementácia aplikácie na správu prezentácií. Táto kapitola zhrňuje postup pri práci a analyzuje, do akej mieri bol splnený cieľ práce, a aký je jej prínos a perspektíva do budúcna.

6.1 Prínos práce

Táto práca splnila moje očakávania, pretože som sa pri nej naučil navrhovať a vyvíjať plnohodnotné aplikácie pre platformu Android. Popri tom som si rozšíril svoje znalosti o súvisiacich technológiách, ktoré sú pri vývoji aplikácii pre mobilné zariadenia využívané a vyžadované aj pri budúcom zamestnaní. Pri písaní tejto práce som sa naučil princípy programovacieho jazyka Java a súvisiacich technológií, najmä jej odlišnosti oproti jazyku *C++*. Jedná sa hlavne o používanie balíčkovacieho systému, vytváranie jar archívov, preklad pomocou nástroja Ant, používanie súborov vo formáte XML. Tieto teoretické znalosti úzko súvisia s platformou Android, ktorá je na jazyku Java postavená. Ďalším prínosom práce bolo zozámenie sa s princípami operačného systému Android a s jeho aplikačným rozhraním. Pri práci som si osvojil prácu s programovým rozhraním Eclipse, na vývoj a testovanie programov pre Android. Zoznámil som sa so základnými konzolovými nástrojmi Android SDK, s vytváraním návrhových XML súborov a kódovaním aplikačných blokov. Pri práci som spoznal princípy SQLite databázy, ktorú využíva väčšina aplikácií napísaných pre Android, ale aj pre iné mobilné platformy. Touto prácou som sa naučil samostatne vyhľadávať a využívať nové zdroje informácií, a konzultovať a prezentovať výsledky svojej práce.

Prínosom práce je aj samotná aplikácia na správu prezentácií, ktorá umožňuje organizovať bloky prezentácií a dodržiavať časový harmonogram pri prezentovaní. Pri analýze požiadavkov a testovaní praktickej použiteľnosti boli navrhnuté jej možné rošírenia do budúcna. Táto technická správa a priložený program môže byť využitý aj na štúdium platformy Android a praktických aspektov pri vývoji Android aplikácií.

6.2 Budúci vývoj

Pri testovaní aplikácie z hľadiska praktickej použiteľnosti sa zistili niektoré nedostatky aplikácie a navrhli sa rozšírenia do budúcna. Aplikácia by mohla v budúcom vývoji ešte viacej prepracovať užívateľské rozhranie najmä z hľadiska estetičnosti, časovej efektivity(užívateľské rozhranie navrhnúť tak, aby bola práca s ním čo najrýchlejšia) a optima-

lizácie na zariadenia s rôznymi rozlíšeniami obrazovky. Ďalším krokom by mal byť návrh a implementácia zistených rozšírení, ktoré používatelia vyžadujú. Rozšírením môže byť napríklad implementácia jednoduchého kalendára, ktorý bude prehľadne zobrazovať naplánované akcie. Aplikácia môže taktiež prepracovať organizáciu obhajob, a rozšíriť možnosti upozornení na blížiaci sa koniec obhajoby. Pri prezentáciách je vhodné prepracovanie možností pre nácvik prezentácie a umožnenie viacerých nastavení. Aplikácia môže povolovať vkladanie a kreslenie obrázkov k jednotlivým snímkom a nahrávať zvuk pre spätnú kontrolu pre užívateľa. Môže zaisťovať synchronizáciu snímkov s PC a priblížiť sa tak kvalite a možnostiam dostupných komerčných aplikácií na správu prezentácií. Program by mal byť umiestnený na Android Market, aby bol pre užívateľov jednoducho dostupný na stiahnutie a aby mala aplikácia cennú spätnú odozvu na ďalšie iteratívne vylepšenia.

Literatura

[1] Google: Google Play. [online], 2012.
URL https://play.google.com/store

[2] Google: Android Developers *Activities*. [online], 2012-05-09 [cit. 2012-05-13].
URL
http://developer.android.com/guide/topics/fundamentals/activities.html

[3] Google: Android Developers *Application Fundamentals*. [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/topics/fundamentals.html

[4] Google: Android Developers *Application Resources*. [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/topics/resources/index.html

[5] Google: Android Developers *Data Storage*. [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/topics/data/data-storage.html

[6] Google: Android Developers *Intents and Intent Filters*. [online], 2012-05-09 [cit. 2012-05-13].
URL
http://developer.android.com/guide/topics/intents/intents-filters.html

[7] Google: Android Developers *Processes and Threads*. [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/topics/fundamentals/processes-and-threads.html

[8] Google: Android Developers *The AndroidManifest.xml File*. [online], 2012-05-09 [cit. 2012-05-13].
URL
http://developer.android.com/guide/topics/manifest/manifest-intro.html

[9] Google: Android Developers *User Interface*. [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/topics/ui/index.html

[10] Google: Android Developers *What is Android?* [online], 2012-05-09 [cit. 2012-05-13].
URL http://developer.android.com/guide/basics/what-is-android.html

[11] Herout, P.: *Java a XML: pro Javu 5 i 6*. Kopp, 2007, iSBN 978-80-7232-307-4.

[12] Murphy, M. L.: *Android 2: Průvodce programováním mobilních aplikací*. ComputerPress, 2011, iSBN 978-80-251-3194-7.

[13] Samy, M.: Using SQLite Database with Android. [online], 2011-05-12 [cit. 2012-05-13].
URL http://www.codeproject.com/Articles/119293/
Using-SQLite-Database-with-Android

[14] Wikipedia: Open Handset Alliance. [online], 2012.
URL http://en.wikipedia.org/wiki/Open_Handset_Alliance

[15] Wikipedia: Android Developers *Services*. [online], 2012-05-09 [cit. 2012-05-13].
URL
http://developer.android.com/guide/topics/fundamentals/services.html

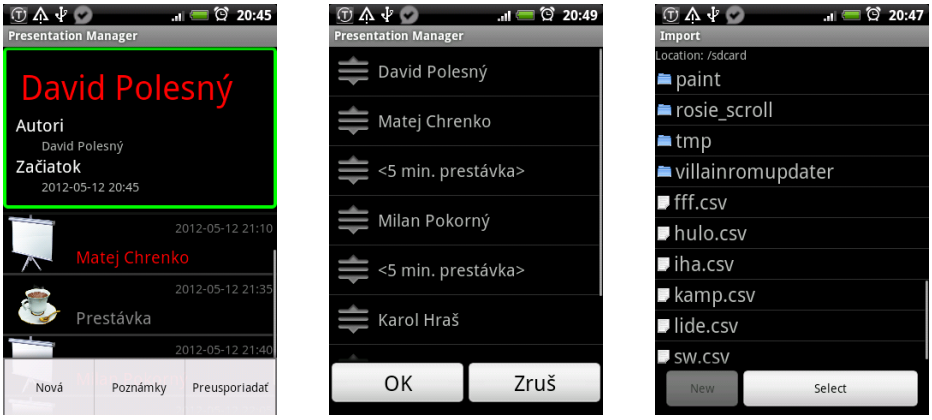
[16] Wikipedia: Android (operating system). [online], 2012-05-12 [cit. 2012-05-13].
URL http://en.wikipedia.org/wiki/Android_(operating_system)

[17] Zakhour, S.: *Java 6: Výukový kurz*. ComputerPress, 2007, iSBN 978-80-251-1575-6.

Příloha A

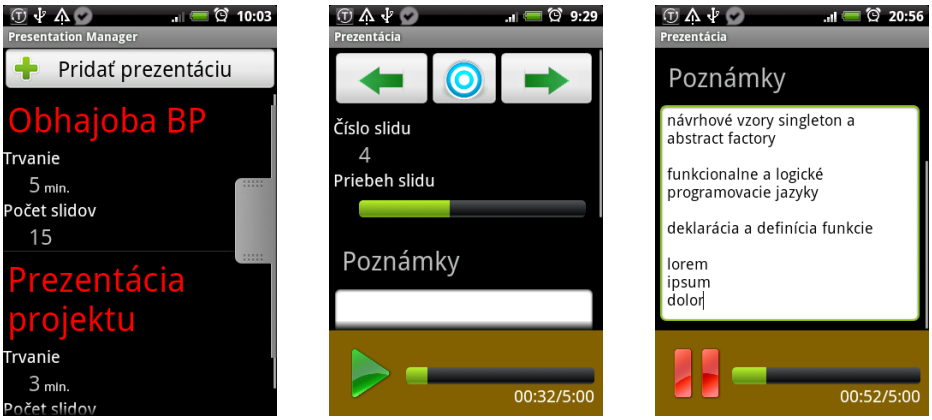
Obrazovky aplikácie

Obrázek A.1: Obrazovka na prepínanie módov, obrazovka akcií s vysunutým menu, dialógové okno na pridanie a editáciu akcie

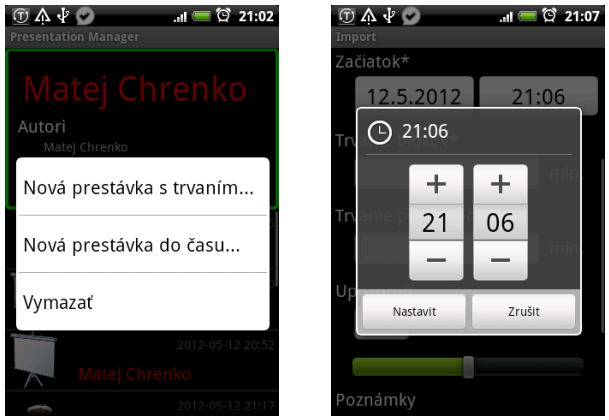


Obrázek A.2: Obrazovka prezentácií s práve prebiehajúcou prezentáciou, okno na preusporiadavanie prezentácií, okno na výber súboru pri importe

Obrázek A.3: Obrazovka aktuálne prebiehajúcej prezentácie, okno na editáciu poznámok, obrazovka nastavení



Obrázek A.4: Obrazovka prezentácií, okno pomoci pri prezentácii, posunuté okno pomoci pri prezentácii



Obrázek A.5: Ukážka kontextového menu, ukážka dialógu na nastavenie času

Příloha B

Dotazník na zisťovanie praktickej použiteľnosti aplikácie

Úloha 1: Spustenie aplikácie a zoznámenie sa s ňou

Spustite aplikáciu, a bez dodatočných informácií sa pokúste odhadnúť na čo slúži, a snažte sa zistiť ako sa ovláda. (Na túto úlohu sú určené maximálne 2 minúty času.)

Úloha 2: Vytvorenie akcie

Vytvorte novú akciu, ktorá začne prebiehať o minútu po vložení a pridajte do nej 2 prednášky. (Povinné parametre zadajte ľubovoľne.)

Úloha 3: Práca s aktuálnou prezentáciou

V nastaveniach aplikácie (v hlavnom okne aplikácie) povoľte používanie bodov, zobrazte aktuálnu prezentáciu vo vytvorenej akcii (je v zelenom rámiku), spustite jej priebeh a zadajte k prezentácii ľubovoľné hodnotenie.

Úloha 4: Vyskúšanie módu pre prezentujúcich

Prepnite sa do módu pre prezentujúcich, vytvorte novú prezentáciu s minimálne dvoma snímkami a vložte k nim poznámky.

Otázka 1: Pomohla by vám pri zadaných úlohách nápoveda v aplikácii?

A) Áno, výrazne rýchlejšie by som splnil zadané úlohy.
B) Áno, ale aplikácia je dosť intuitívna aj bez nej.
C) Nie, vôbec nie je potrebná pri takto jednoduchej aplikácii.

Otázka 2: S ktorými problémami ste sa už stretli pri prezentácii alebo obhajobe projektu?

A) Nedodržal som stanovený čas na prezentáciu.
B) Zabudol som odprezentovať niektorý z pripravených bodov prezentácie.
C) Nestihol som sa pripravil na prezentáciu.

Otázka 3: Využili by ste túto aplikáciu pri svojej prezentácii?

A) Áno, keby som nenašiel na Android Markete lepšiu aplikáciu.
B) Nie, aplikácia by mi nepomohla.

Otázka 4: Čo by ste na aplikácii doplnili alebo vylepšili?

Příloha C

Obsah CD

CD priložené k technickej správe obsahuje v koreňovom adresári tieto položky:

- `TechnicalReport.pdf` - technická správa v elektronickej podobe
- `tex-sources/` - zložka so zdrojovými kódmi technickej správy
- `PresentationManager.apk` - inštalačný súbor aplikácie
- `apk-sources/` - zložka so zdrojovými kódmi aplikácie