



ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

SEMESTRÁLNÍ PRÁCE KIV/MKZ
Studentův pomocník

Jaroslav Klaus
24. května 2016, Plzeň

Obsah

1	Zadání	2
2	Programátorská dokumentace	2
2.1	Uložení dat	2
2.2	Hlavní obrazovka	2
2.3	Přidání rozvrhu	3
2.4	Odebrání rozvrhu	3
2.5	Nastavení	3
2.6	Zobrazení sylabu	3
2.7	Podmínky	4
3	Uživatelská dokumentace	4
4	Řešené problémy	4
5	Testování	4
6	Závěr	4

1 Zadání

V rámci samostatné práce je potřeba vytvořit předem odsouhlasenou aplikaci na zvolené cílové platformě, předvést funkčnost aplikace v emulátoru nebo na fyzickém zařízení a odevzdat okomentovaný kód práce včetně dokumentace.

Pro splnění těchto podmínek jsem si zvolil tvorbu aplikace pro platformu Android, která by měla sloužit jako pomocník pro studenta vysoké školy, na které je používán informační systém Stag. Aplikace bude zobrazovat rozvrh studenta získaný ze Stagu po dnech a dovolí mu si ke každému z předmětů zobrazit či přidat další informace, jako jsou

- sylabus předmětu,
- podmínky pro splnění předmětu,
- úkoly, které je potřeba udělat
- a počet absencí v předmětu.

2 Programátorská dokumentace

2.1 Uložení dat

Data jako rozvrhy a sylaby předmětů jsou uloženy v XML souborech ve složkách strukturou vycházející z webové adresy, ze které byly získány, a osobního čísla rozvrhu. Pro jejich čtení a parsování byla vytvořena třída *ParseXmls*, která využívá třídu *XmlPullParser*.

Konfigurační položky a záznamy o tom, jaké rozvrhy jsou dostupné, jsou uloženy ve sdílených preferenčních souborech. To jsou typicky také XML soubory, ale o jejich čtení a ukládání se stará poskytované Android API.

Data o podmínkách a úkolech se uchovávají v databázi v příslušných tabulkách. K manipulaci s nimi jsou vytvořené třídy *TasksDatabaseHelper* a *TermsDatabaseHelper* oddělené od *SQLiteOpenHelper*.

Veškerá práce s daty a jiné eventuálně časově náročné akce jsou prováděny v jiných vláknech pomocí třídy *AsyncTask*.

2.2 Hlavní obrazovka

Hlavní obrazovka *MainActivity* zobrazuje předměty po jednotlivých dnech, proto byla implementována pomocí *Tabbed Activity* s volbou *Action Bar Tabs* (with *ViewPager*). Základem jsou tedy „záložky“ se dny zobrazující

fragmenty, o jejichž změnu se stará třída *ViewPager* a její adaptér oddělený od třídy *FragmentStatePagerAdapter*.

Tento adaptér vytváří fragment *DayFragment*, který v prvku *ExpandableListView* zobrazuje rozvrh pro daný den. Pro to je však potřeba další adaptér (*BaseExpandableListAdapter*), který je použit pro zobrazení údajů o předmětu v prvku *ExpandableListView*.

Při vytvoření aplikace se z konfiguračních souborů zjistí poslední nastavení a podle toho se načte příslušný rozvrh do struktur. Poté se naplní prvek *Spinner* dostupnými staženými rozvrhy. Tento prvek slouží ke změně zobrazovaného rozvrhu.

2.3 Přidání rozvrhu

Aktivita pro přidání rozvrhu *AddTimetableActivity* je základní aktivita vytvořená jako *Basic Activity*. Obsahuje tři prvky: *Spinner*, *EditText* a *Button*.

Spinner slouží pro výběr školy a je pro něj vytvořen adaptér pomocí *ArrayAdapter<String>*, který umožňuje nejen zobrazení názvu školy, ale i jejího loga.

EditText je vstupní pole sloužící pro zadání osobního čísla studenta, jehož rozvrh má být přidán.

Button je tlačítko, které slouží pro zahájení akce přidání nového rozvrhu.

Pro stahování souborů se využívá tříd *URL*, *URLConnection*, *BufferedInputStream* a *FileOutputStream*.

2.4 Odebrání rozvrhu

Pro odebrání rozvrhů se využívá třídy *AlertDialog.Builder*, pomocí které se vytvoří dialog se zatrhávacími políčky u osobních čísel (metodou *setMultiChoiceItems*). Vybrané položky se poté vymažou.

2.5 Nastavení

Aktivita nastavení *SettingsActivity* obsahuje pouze dvě položky, a to pro změnu osobního čísla a semestru. Obě položky jsou realizovány pomocí prvku *ListPreference*.

2.6 Zobrazení sylabu

Aktivita pro zobrazení sylabu předmětu je vytvořena jako *Basic Activity*. Obsahuje pouze prvky *TextView*, které slouží pro zobrazení příslušných informací o předmětu.

2.7 Podmínky

Aktivita pro zobrazení podmínek *TasksActivity* je upravená *Basic Activity* která v prvku *ListView* zobrazuje podmínky načtené z databáze. Využívá k tomu adaptér typu *ArrayAdapter<Term>*, který slouží k vytvoření a nastavení hodnot zobrazovaných prvků.

Pro vytvoření nového prvku je použito tlačítko *FloatingActionButton*, které spustí aktivitu *AddTermActivity*. Stejná aktivita se spustí pro úpravu existující podmínky a to tak, že se na danou podmínku klikne.

3 Uživatelská dokumentace

4 Řešené problémy

5 Testování

6 Závěr

Reference