

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Informační portál pro studenty ZČU na platformě Android

Plzeň, 2013

Jan Bláha

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 8. srpna 2013

Jan Bláha

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Ladislavu Pešíčkovi především za jeho ochotu a trpělivost. Rád bych také poděkoval své rodině za veškerou poskytnutou podporu.

Abstract

The purpose of this work is to develop interactive application working on Android platform. This application will provide students of University of West Bohemia information about their faculties, location and other useful information. In the first part I try to show why did I choose Android platform over others and introduction of information servers of University. The second part is compassion and introduction of each activity. Product of this task is a working application which provides data from selected information sources and runs on a real device.

Obsah

1 Úvod.....	1
2 Cíl práce	2
3 Platformy mobilních zařízení	3
3.1 Android	3
3.2 Ostatní platformy	5
3.3 Výběr platformy a verze API	6
3.4 Vývojové prostředí a požadavky	7
3.5 Použité senzory	8
4 Podobné aplikace	11
4.1 Aplikace Harvard University	11
4.2 Aplikace Princeton University	13
4.3 Další podobné aplikace	13
5 Analýza problému	14
5.1 Informační servery ZČU	14
5.2 Zpracování vybraných informací	17
5.2.1 Přímý přístup	17
5.2.2 Nepřímý přístup	18
6 Návrh aplikace	19
6.1 Aktivita Zcu_Info	21
6.2 Aktivita Maps	23
6.2.1 Vrstvy	25
6.2.2 Offline mapy	26
6.3 Aktivita Menza	27
6.4 Aktivita Novinky	28
6.5 Aktivita Udalosti	29

6.6	Preference uživatele	30
6.7	Aktivita Pomoc	31
6.8	Full Web.....	31
6.9	Aktivita Aktualizace	32
6.10	Menu	35
7	Testování aplikace.....	37
7.1	Běh aplikace	38
7.2	Správnost Dat.....	38
7.3	Určování polohy.....	40
7.4	Jednání dle preferencí	40
8	Možnosti rozšíření aplikace	41
9	Závěr	43

1 Úvod

Mobilní telefony se již dávno staly součástí našich životů a málokdo si dokáže představit život bez nich. Poslední dobou je již spíše výjimka nevlastnit takzvaný chytrý telefon, který nám kromě telefonování a psaní krátkých zpráv poskytuje velké spektrum dalších možností. S rozšířením těchto zařízení mezi veřejnost, začali i programátoři ve vývoji širokého spektra aplikací.

I přes možnost zobrazovat webové stránky, na zařízeních podporující připojení k internetu, je složité orientovat se ve všech informačních zdrojích. Tento fakt podporuje i malá velikost mobilních zařízení, která málokdy přesahuje 5 palců. Proto se spousta poskytovatelů obsahu uchyluje k tvorbě aplikací, které by ulehčily orientaci ve velkém množství informací.

Svojí prací chci poskytnout užitečné informace studentům Západočeské univerzity vlastními zařízeními, které pracuje na systému Android. Patří mezi ně například snadnější orientace v areálu Západočeské univerzity, což využijí především studenti, kteří nepocházejí z Plzně a kterým je prostředí zcela nové. Nejenom oni však mohou být uživateli, protože aplikace bude nabízet i průvodce v systému pojmenování místností na celé ZČU. V aplikaci počítám s možností připojení k internetu, které je v dnešní době dostupné téměř všude.

Studenti se budou moci zúčastnit akcí pořádanými univerzitou, o kterých se díky zobrazování informací o událostech a novinek z prostředí univerzity dozví.

Doufám, že aplikace bude v budoucnu využívána studenty ZČU.

2 Cíl práce

Primárním cílem mé práce je vytvořit aplikaci, která by bez připojení k internetu dokázala pomoci studentovi v orientaci v rozsáhlém areálu ZČU pomocí geografických funkcí mobilního zařízení. V případě nedostupnosti geografických funkcí bude aplikace schopná ukázat na mapě polohu hledané budovy, nebo místnosti. Dále bude poskytovat základní informace o ZČU a jejích objektech. Především pak o studijních odděleních jednotlivých fakult a jejich kontaktech.

Na vyžádání poskytované aktualizace aplikace budou sloužit k aktualizaci již přítomných souborů a databází, ze kterých aplikace čerpá všechna svá data.

Připojení k internetu také slouží k vyvolání služby Google Maps, která ukáže uživateli jak jeho aktuální polohu, tak polohu všech budov ZČU v Plzni. K těmto bodům zájmu bude připojena i stručná informace o tom, jaká fakulta zde sídlí a jaké místnosti (pouze kód místnosti složený ze dvou písmen) tento bod zájmu obsahuje.

3 Platformy mobilních zařízení

V této kapitole se věnuji porovnání jednotlivých platforem mobilních zařízení. Větší část kapitoly je věnována platformě Android a jejímu podrobnějšímu prozkoumání, jakožto cílové platformě této práce. Mezi porovnávané platformy jsem řadil pouze ty, které mají více než 3% zastoupení na světovém trhu. [3]

3.1 Android

Operační systém Android je založený na Linuxovém jádru a navržený speciálně pro mobilní zařízení, jako jsou chytré telefony, tablety a počítače s dotykovým displejem. Celý systém je Open source vydávaný pod Apache License, což umožňuje, aby byl tento software volně využíván a modifikován od výrobců mobilních zařízení až po samotné uživatele.

Android je vyvíjen společností Open Handset Alliance, která je vlastněna firmou Google. Android je nyní nejlépe prodávanou platformou a každý den je aktivováno více než milion nových zařízení běžících na systému Android. [1]

Operační systém Android byl uveden na trh koncem roku 2008 na zařízení T – Mobile G1.[10] Verze Application Programming Interface (dále jen API) byla společností Open Handset Alliance určena jako úroveň 1.

V současné době již byla uvedena verze Androidu 4.3 nesoucí kódové označení, stejné jako předchozí verze, Jelly Bean. API úroveň 18 značí, že od prvního Androidu již bylo na trh uvedeno 17 nových úrovní. Mezi nové vlastnosti této verze patří například inteligentní našeptávání telefonních čísel, bezdrátové propojení s televizí, nebo možnost vytvoření profilů více uživatelů zařízení.[14]

Verze	Kódové označení	API	Podíl na trhu
1.6	Donut	4	0.1%
2.1	Eclair	7	1.4%
2.2	Froyo	8	3.1%
2.3.3 - 2.3.7	Gingerbread	10	34.1%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	23.3%
4.1.x	Jelly Bean	16	32.3%
4.2.x	Jelly Bean	17	5.6%

Tabulka 3-1 Podíl jednotlivých Android zařízení v provozu v závislosti na verzi API

Dnes je stále nejpoužívanější Android 2.3.3 – 2.3.7 s kódovým jménem Gingerbread. Používá ho více jak 34% uživatelů (viz. Tabulka 2.1) operačního systému Android [2].

Koncem roku 2012 více než 70% všech mobilních zařízení fungovalo právě na systému Android. [1]

Pokud chceme pro naši aplikaci fungující na systému Android co největší potenciál je nutná minimální verze API úroveň 10 Gingerbread. Pokud bychom zvolili větší úroveň API, přijdeme až o třetinu potenciálních uživatelů.

3.2 Ostatní platformy

Mezi hlavní konkurenty Androidu patří: Symbian, BlackBerry, iOS, Windows Phone.

Symbian OS

Po ukončení spolupráce firmy Nokia a Symbian Foudation, v roce 2011, se stala budoucnost Symbian OS nejistou. V současné době není tento systém přímo podporován žádným výrobcem. I přesto měl Symbian v roce 2012 stále více než 3% zastoupení na světovém trhu s mobilními zařízeními. [3] Za poslední rok bylo aktivováno 23,9 milionů zařízení s tímto operačním systémem. [3]

BlackBerry OS

Tento operační systém byl vytvořen kanadskou firmou Research in Motion (RIM) a je používán výhradně na zařízeních značky BlackBerry. Mezi ostatními platformami zaujme možnost programovat aplikace pro tento systém v 5 programovacích jazycích. BlackBerry nám také nabízí možnost vzít již vytvořenou aplikaci pro Android a zkonvertovat ji právě pro systém BlackBerry. [4] Se 4,5% byl tento systém třetím nejpoužívanějším v roce 2012. [3]

iOS

iOS je platforma vytvořená pro potřeby jedné firmy a tou je firma Apple. iOS je, stejně jako MacOS X, založen na UNIXu. Vývoj aplikací probíhá v prostředí Xcode, které je dostupné pouze pro Mac OS X. [5] Přes omezení pouze na výrobky iPad, iPod Touch a iPhone je iOS stále platformou s největším počtem nabízených aplikací v oficiálních internetových obchodech.[5] Vývoj probíhá v jazyce Objective C, který je nadstavbou jazyka C. V roce 2012 měl iOS zastoupení na trhu 18,8%. [3]

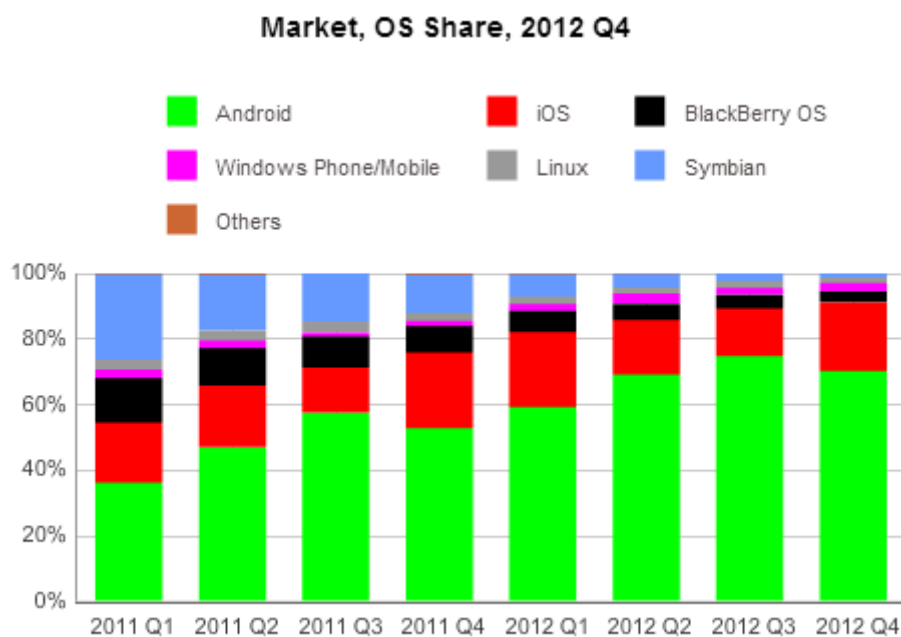
Windows Phone

Systém Windows Phone 7 vznikl jako pokračování systému Windows Mobile. Avšak není s tímto systémem kompatibilní. V roce 2012 vyšla verze Windows Phone 8. Již Windows Phone 7 představilo prostředí Modern UI, které bylo použito i pro OS Windows 8. Pro vývoj aplikací poskytuje Microsoft zdarma množství nástrojů. Vývojové prostředí Visual Studio 2010 Express for Windows Phone a emulátor pro testování aplikací Windows phone Emulator patří mezi nejdůležitější. [6]

3.3 Výběr platformy a verze API

Výběr platformy pro tvorbu aplikace značně ovlivnila současná situace a rozložení na trhu s mobilními operačními systémy. Android na tomto trhu již od druhé poloviny roku 2011 zastává více než 50% podíl. [3]

Zvolil jsem nejnižší úroveň API Level 8, Androidu s kódovým označením Froyo z důvodu jejího stálého využití (viz. Tabulka 2-1). Toto využití nadále klesá s počtem nových Android zařízení (v lednu 2013 se pohybovalo okolo 7% [2]), ale v současné situaci je její zastoupení stále nezanedbatelné. Především se jedná o uživatele, kteří mají starší „chytrý“ telefon, jako je například HTC Desire, HTC Wildfire, nebo Samsung Galaxy S.



Obrázek 3-1 Rozložení trhu s mobilními telefony v závislosti na OS ve 4. Kvartálu 2012

3.4 Vývojové prostředí a požadavky

Pro vývoj aplikací fungujících na Androidu je využíván především jazyk Java. Na stránkách [1] je zdarma k dispozici Android SDK, který obsahuje nástroje a API potřebné pro vývoj aplikací na platformě Android. Jako preferované vývojové prostředí jsem zvolil Eclipse IDE, které je od verze 3.6 (Helios) kompatibilní s pluginem ADT [7].

Požadavky Android SDK na operační systémy jsou následující:

- Windows XP (32-bit), Vista (32- nebo 64-bit), Windows 7 (32- nebo 64-bit), nebo Windows 8
- Mac OS X 10.5.8 nebo novější (pouze x86)
- Linux (testováno na Ubuntu Linux, Lucid Lynx)
 - GNU C knihovny (glibc 2.7, nebo novější)
 - Pro Ubuntu Linux verze 8.04, nebo novější
 - 64-bit distribuce musí být schopné spustit 32-bit aplikaci

Další vývojové prostředí

Mezi vývojová prostředí nedávno Google přidal prostředí Android Studio. Jedná se o prostředí podobné Eclipse s ADT Pluginem založené na IntelliJ IDEA. Android Studio nabízí vývojáři množství nástrojů pro vývoj a ladění programů. Android Studio je v současné době nabízeno jako preview.[15]

3.5 Použité senzory

Aplikace využívá několika senzorů, které jsou přítomny v Android zařízení. Ty nám poskytují nejenom informace o naší poloze, ale i informace o stavu zařízení. Díky senzorům dokáže zařízení například rozpoznat vzdálenost zařízení od objektu v těsné blízkosti zařízení a v závislosti na této informaci, například, snížit jas displeje během hovorů. Android podporuje tři hlavní kategorie senzorů. Jsou to:

- **Pohybové senzory**

Tyto senzory měří rotační a zrychlovací síly kolem tří os. Tato kategorie obsahuje akcelerometry, gyroskopy a gravitační senzory.

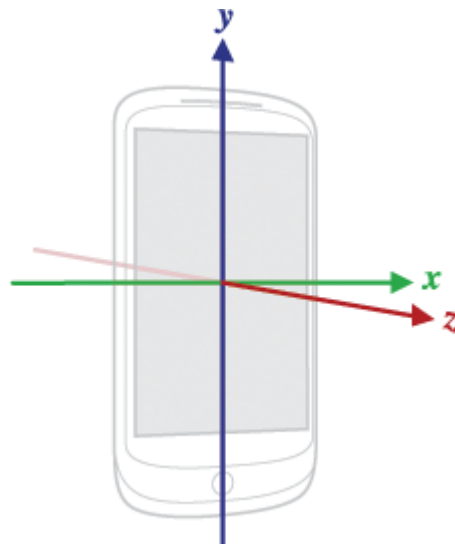
- **Senzory vnímání okolního prostředí**

Tyto senzory měří vlastnosti prostředí v blízkém okolí telefonu. Jedná se například o teplotu vzduchu, tlak vzduchu, míru osvětlení a vlhkost.

- **Poziční senzory**

Tyto senzory měří fyzickou polohu zařízení. Jedná se o orientační senzory a magnetometry.

Pro moji aplikaci jsem použil pouze pohybové a poziční senzory. Z pohybových se jedná o gravitační sensor. Ten byl použit z důvodu otočení zařízení o 90° a změny obrazovky. Z pozičních senzorů jsem použil orientační sensor. Tento sensor umožňuje monitorovat pozici zařízení na zemském povrchu vzhledem



Obrázek 3-2 Nastavení jednotlivých os v Android mobilním zařízení. Převezato z [13]

k severnímu magnetickému pólu a využívám ho jako pomocný senzor při určování polohy pomocí GPS senzoru.

Orientační senzor pracuje se třemi na sobě nezávislými hodnotami. Jedná se o hodnoty Azimuth, Pitch a Roll, které měří pohyb zařízení vzhledem k severnímu magnetickému pólu podle os z, x a y v tomto pořadí.

Následující příklad zdrojového kódu ukazuje, jak můžeme získat data o orientaci přímo z orientačního senzoru. Kód příkladu je převzat z [13].

Při zavolání aktivity dojde k propojení výstupu senzoru s námi vytvořenou proměnnou `mSensorManager`. Také upřesníme, že se jedná o senzor orientační.

```
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);
mOrientation =
mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION);
```

Protože každá aktivita musí počítat s neplánovaným ukončením, pomůže nám metoda `onPause()`, která v případě odsunutí aktivity na pozadí přestane sbírat data ze senzoru a tím i uvolní zabranou paměť.

```
@Override
protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);}
```

Podobně jako nám metoda `onPause()` pomáhá při odsunutí aktivity na pozadí, pomáhá metoda `onResume()` s opětovným načítáním zdrojů v případě, že se uživatel opět začne aktivitě věnovat. Díky tomu nemusíme znovu spouštět celou aktivitu, ale pouze znovu zapneme sběr dat a můžeme pokračovat na stejném místě, jako když byla aktivita odsunuta na pozadí.

```
@Override
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this, mOrientation,
    SensorManager.SENSOR_DELAY_NORMAL);}
```

Metoda, která pracuje s daty získanými přímo od senzoru. Do proměnné typu float ukládá jednotlivé změny.

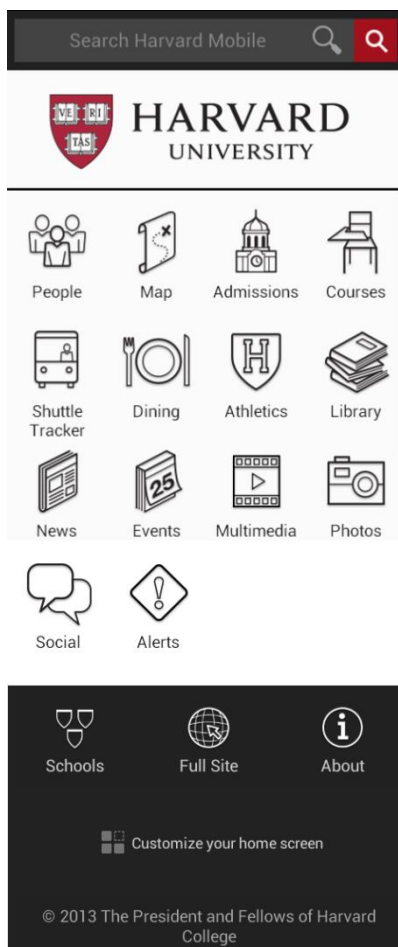
```
@Override
public void onSensorChanged(SensorEvent event) {
    float azimuth_angle = event.values[0];
    float pitch_angle = event.values[1];
    float roll_angle = event.values[2];
    // zde můžeme pracovat se získanými daty
}
```


4 Podobné aplikace

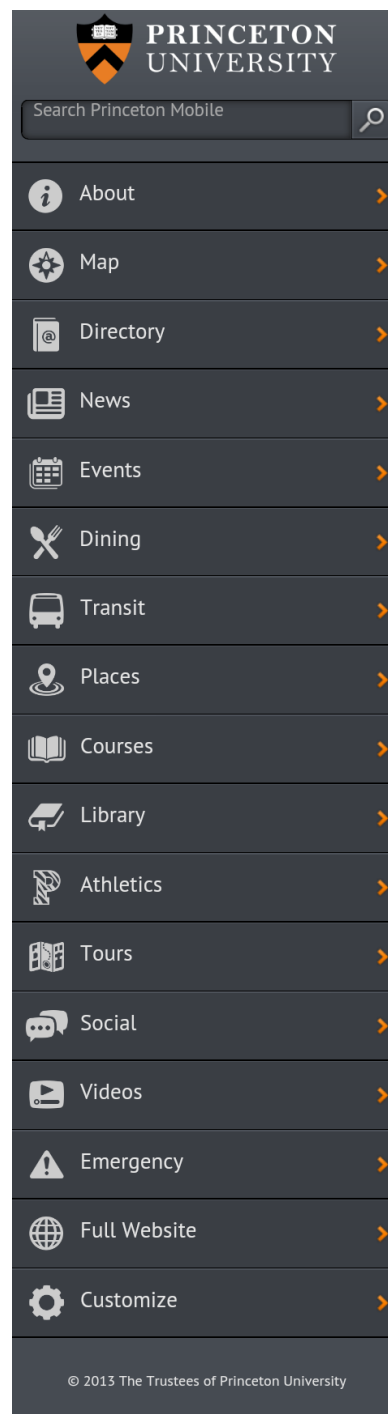
Během mé práce jsem prozkoumal na Google Play několik aplikací, které se zabývali podobnou tematikou jako moje práce. Vybral jsem dvě nejzajímavější z nich a ty jsem dále analyzoval. Obě tyto aplikace jsou pro zařízení fungující na systému Android.

4.1 Aplikace Harvard University

Aplikace běží na prohlížeči založeném na jádře WebKit, které zobrazuje konkrétní URL adresu. Od ostatních webových prohlížečů se liší především tím, že neobsahuje žádné navigační prvky a umožňuje nám interakci pouze se zobrazenou stránkou. V případě jakékoliv interakce uživatele s aplikací, má vývojář právo předem určit, jak se aplikace zachová. Tato aplikace využívá upravené webové stránky, pro mobilní zařízení, pro poskytování všech informací a pracuje, pouze tedy když je uživatel připojen k internetu. Aplikace pracuje s lokačními a GPS senzory pro zobrazení aktuální polohy uživatele v objektu univerzity. Aktivita Shuttle Tracker nabízí možnost zjištění dopravního spojení v celém městě. Aktivita Athletics, News a Events nám nabízí možnost načtení aktuálního dění nejenom na kampusu univerzity, ale vše co souvisí s univerzitou. Multimedia nás navedou na Youtube kanál univerzity se všemi jejími videi a Photos na Flickr účet. Harvard University obsahuje také vyhledávání lidí z univerzity, předměty možné k zapsání v abecedním pořadí (viz. Obrázek 4-1).



Obrázek 4-1 Aplikace Harvard University



Obrázek 4-2 Aplikace Princeton University

4.2 Aplikace Princeton University

Aplikace Princeton University slouží jako mobilní průvodce při zjišťování informací o univerzitě v Princetonu. Dokáže vás navádět v areálu univerzity, ukazovat aktuální novinky a události. Označí vám důležitá místa a navede vás po zajímavostech v areálu univerzity atd. (viz Obrázek 4-2). Tato aplikace je založena na vlastním prohlížeči s WebKit jádrem. Je však limitována na neustálé připojení k internetu. Zobrazuje nám upravenou webovou stránku, která je přímo určena na zobrazování na mobilních zařízeních. Je také vybavena posuvným zobrazováním (ScrollView), pomocí něhož můžeme posouvat právě zobrazenou část horizontálním směrem. Aplikace Princeton University nabízí, oproti Harvard University, větší množství funkcí. Kromě již představených funkcí jako je zobrazování polohy, novinek, událostí a všech předmětů univerzity nabízí tato aplikace i předem připravené trasy, které vás provedou celou univerzitou. Tato funkce využívá senzory pro sledování polohy a stavu zařízení, které jí dávají vědět aktuální informace o poloze.

4.3 Další podobné aplikace

Mezi další podobné aplikace, které jsem analyzoval, patří The University of Alabama, Ashford University Mobile, University of Phoenix, Vanderbilt University a Columbia University. Všechny tyto aplikace poskytují velké množství informací pro studenty ohledně dění na konkrétní univerzitě. Některé aplikace poskytují určité množství informací až po přihlášení do studentského systému. Například aplikace Columbia University pracuje, kromě zobrazení mapy, pouze po přihlášení do studentského systému.

5 Analýza problému

První věc, na kterou jsem se zaměřil ještě před tvorbou aplikace, byly informační zdroje poskytované univerzitou a městem Plzeň, které by pro studenty byly nejvhodnější. Jejich důležitost, aktuálnost a informační hodnotu pro studenty ZČU jsem shrnul do několika vět.

5.1 Informační servery ZČU

Při výběru zdrojů ZČU jsem postupoval podle Průvodce prváka 2012 [8], distribuovaným Západočeskou univerzitou. Snažil jsem se nehlédět na důležitost informací z pohledu studenta Fakulty aplikovaných věd, ale studenta obecně, bez preferované fakulty.

ZČU (zcu.cz)

Hlavní komunikační kanál ZČU jako jednotný objekt. Zobrazuje Aktuality a Kalendář akcí celé univerzity – všech fakult. Jednoduchý přístup k domovským stránkám jednotlivých fakult. Spodní menu sloužící k rychlé navigaci při více sofistikovaném požadavku.

Do aplikace jsem použil Aktuality a zobrazení celé stránky univerzity.

Správa kolejí a menz (skm.zcu.cz)

Přehledné informace o kolejích ZČU a kontakty na vedoucí koleje, stejně tak jako adresa koleje. Informace o jednotlivých menzách včetně jídelníčku.

Do aplikace jsem použil informace o kolejích včetně kontaktu na vedoucí těchto kolejí. Také jsem použil možnost zobrazení jídelníčku obou plzeňských menz.

Portál (portal.zcu.cz)

Adresa, na které student tráví velkou spoustu času. Ať se jedná o předzápis, zápis na termín zkoušky, nebo kontrola rozvrhu. Vyžaduje přihlášení pomocí Orion hesla. Nepřihlášený uživatel může zobrazit předměty z celé univerzity a jejich studijní plán nebo vybrat studijní obor i s vizualizací a informacemi o tomto oboru. Na portálu je již k dispozici aplikace Rozvrh, která je i na Google Play, a aplikace Zkoušky pro mobilní telefony. Proto jsem se rozhodl do své aplikace již tyto funkce neimplementovat. Již existuje také aplikace ukazující jídelníček menzy na týden dopředu, i proto jsem zvolil načítání jídelníčku pouze na aktuální den.

Z portálu jsem použil stahování univerzitních událostí, které mi přišli lépe koncipované a důležitostí informací podstatnější než události na stránce univerzity (zcu.cz).

CIV (support.zcu.cz)

Návod na připojení mobilního zařízení, certifikát k připojení na Eduroam. Připojení k e-mailu. Změna Orion hesla. Aktuality o výpadcích provozu sítě.

Do aplikace nebyla ze serveru CIVu použita žádná informace.

Knihovna (knihovna.zcu.cz)

Vyhledávání knih z databáze Univerzitní knihovny. Informace o stavu knihy. Informace o knihovnách a biblioboxech.

Do aplikace nebyly ze stránek knihovny použity žádné informace.

Webové služby poskytované CIVem (<https://stag-ws.zcu.cz/>)

Přes velké množství webových služeb, které jsou poskytovány CIVem, jsem do aplikace žádnou z těchto služeb neimplementoval. Hlavně z důvodu odlišného konceptu aplikace a poskytovaných služeb. Mezi služby, které by v budoucnu byly vhodné využít, se řadí Kalendář a harmonogram akademického roku, Předměty, Studijní programy a Znamky. Kalendář poskytuje harmonogram akademického roku, včetně svátků a případných změn v rozvrhu, které byly způsobené státními svátky. Služba Předměty vrací jak seznam literatury potřebné pro zadaný předmět, tak i ECTS údaje o zvoleném předmětu. Studijní programy vrací detailní informace o

oboru dle jeho identifikačního čísla. Služba Znamky vrací seznam studentů a jejich známek ze zápočtů a zkoušek na zvoleném předmětu.

Studentské organizace

Velké množství studentských organizací, které nabízí informace důležité pouze pro zainteresovanou část studentů. Jako výjimku bych uvedl server Dioné a Rádio Bomba, které se snaží oslovit všechny studenty univerzity. Dioné navíc nabízí množství zajímavých událostí, které však nejsou oficiálními událostmi univerzity a slouží především k navázání kontaktů mezi samotnými studenty.

Sociální sítě

Západočeská univerzita je i na sociální síti Facebook, kde má vlastní profil. Na tomto profilu univerzita informuje nejenom o událostech a novinkách, ale i nahrává studentská videa a fotografie. Dále prezentuje své výrobky a akce. Na sociálních sítích jsou i některé fakulty a katedry. Sdílené informace jsou však cílené především na studenty těchto fakult a kateder.

5.2 Zpracování vybraných informací

V aplikaci budu přistupovat k předem známým informacím (ve známém formátu) na předem určených webových stránkách (informačních serverech ZČU). Proto si mohu dovolit ke každé stránce přistupovat individuálně a s vědomím, že jejich struktura nebude každý měsíc radikálně změněna.

Přesto, že v současné době je většina mobilních zařízení dobře vybavena, musel jsem stejně volit optimální postup z důvodu menšího úložiště a náročnosti na použité zařízení. Pro co nejlepší přehlednost jsem bral v úvahu také vizuální stránku aplikace.

5.2.1 Přímý přístup

Dalším úkolem bylo rozhodnout, jak přistupovat k datům, která nebudou přímo součástí aplikace. Jedná se především o informace, které budou získávány prostřednictvím internetu. Nabízejí se nám dvě možnosti. První z nich je přímý přístup ke každému dodavateli informací (v tomto případě webovým stránkám) zvlášť. Nevýhodou tohoto přístupu může být velké množství informací, které se bude stahovat najednou z několika zdrojů při každém spuštění nebo aktualizaci. Zdroje navíc musí stále existovat a poskytovat informace ve stejné struktuře jako dosud. To může být velký problém, pokud by poskytovatel informací obměnil strukturu svých webových stránek. I přidání nového zdroje, s jinou strukturou, by vyžadovalo aktualizaci celé aplikace, nejenom vizuálního prostředí. Tyto nevýhody však můžeme ošetřit tím, že aplikace bude fungovat převážně bez připojení k internetu a všechny stažené informace si bude ukládat do interních souborů. Tyto soubory bude aktualizovat na uživatelský pokyn, či podle zadaných preferencí aktualizace jenom některých zdrojů.

5.2.2 Nepřímý přístup

Jako druhá možnost se nám nabízí přístup nepřímý. Tento přístup je založen na síťové komunikaci se serverem, který by fungoval jako prostředník mezi poskytovateli dat a naší aplikací. Nezávisle by zpracovával data a poté na žádost poslal už jenom námi požadované informace v námi známém formátu.

Mezi další přednosti tohoto přístupu můžeme zařadit menší zatížení mobilního zařízení, protože místo nutnosti data stahovat, zpracovat a poté zobrazit, nám bude stačit data pouze stáhnout a zobrazit, což značně ulehčí zařízení. Také dokážeme aplikaci značně zmenšit, protože se bude starat o komunikaci pouze s jedním subjektem.

Na serveru ovšem může být uloženo větší množství dat, proto musíme postupovat u každého informačního zdroje v jasné hierarchii, a stahovat pokaždé jenom data potřebná pro náš výběr.

Z důvodu, že aplikaci chci ladit především do offline prostředí (bez přístupu k internetu) a k tomu, že v online módu budu informace stahovat pouze ze tří zdrojů a to nezávisle na sobě, rozhodl jsem se použít přímý přístup k požadovaným informacím na stránkách jejich poskytovatelů a tyto informace si sám upravit do požadovaného formátu. Zároveň všechna tato data budu ukládat do souborů v aplikaci, které budou sloužit primárně k obsluze aplikace v momentech, kdy připojení k internetu nelze navázat.

6 Návrh aplikace

Pro aplikaci jsem zvolil sadu ikon [9], kterou jsem upravil použitím titulku pro větší přehlednost a srozumitelnost.

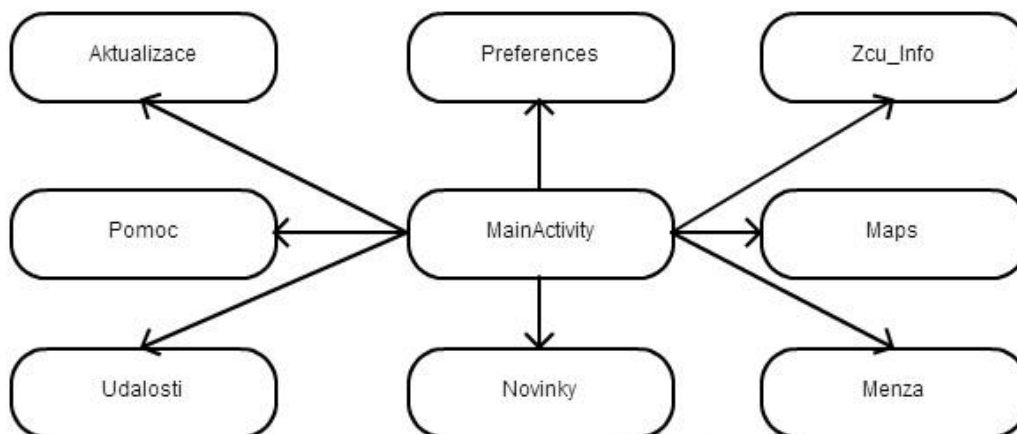
Prostředí Eclipse umožňuje jak grafický návrh, tak i přímou úpravu vygenerovaného XML souboru.

Jako layout hlavní obrazovky, která se načte po zapnutí aplikace, jsem zvolil Table Layout, pomocí něhož jsem mohl vkládat prvky do jednotlivých řádek a nemusel jsem se starat o jejich interní vztahy. Layouty jsou řešeny XML souborem, který specifikuje jednotlivé widgety a jejich základní vlastnosti. Tyto vlastnosti jde měnit i v .java souborech a používá se pro dynamičtější změnu prostředí či pro změnu textu dle výběru uživatele.

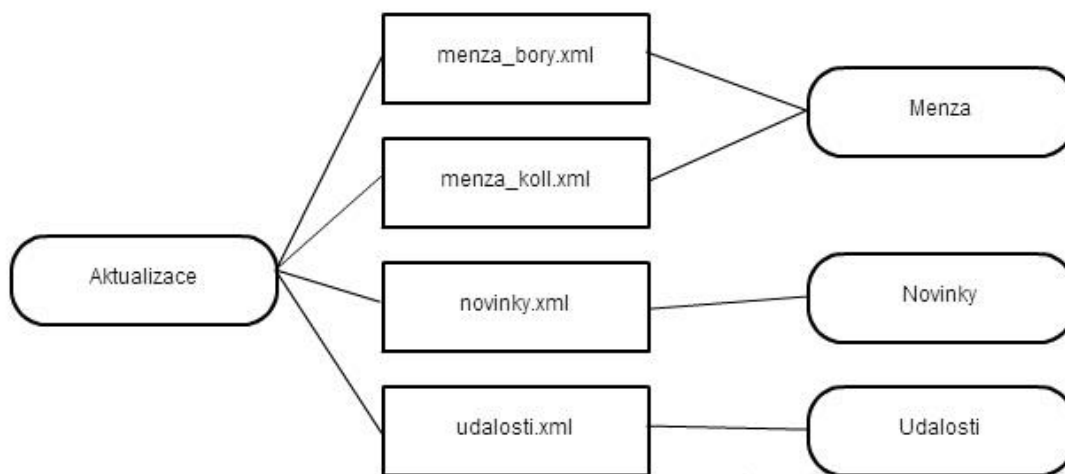


Obrázek 6-1 Úvodní obrazovka aplikace ZČU

Příklad použití Table Layout v souboru aktivity_main.xml najdeme v projektu ve složce res/layout. Na obrázku 6-2 vidíme návaznost jednotlivých aktivit. Je patrné, že všechny aktivity se spouštějí z MainActivity.



Obrázek 6-2 Návaznost všech aktivit v aplikaci



Obrázek6-3 Sdílení souborů aktivitami

Na obrázku 6-3 vidíme, které aktivity pracují se stejnými soubory. Jedná se o XML soubory s námi vytvořenou strukturou (o této struktuře je více napsáno v kapitole 6.3 Aktivita Menza). U aktivity Aktualizace se jedná o zapisování nových dat a vytváření struktury těchto souborů. Naopak u aktivit Menza, Novinky a Udalosti se jedná o čtení informací z těchto souborů a na základě těchto souborů vytvářejí vlastní výčty. Soubory se ukládají do interního úložiště přímo do zařízení a může k nim přistupovat pouze naše aplikace.

6.1 Aktivita Zcu_Info

Základním komunikačním prvkem mezi jednotlivými aktivitami v aplikaci pro Android je záměr (Intent). Tento záměr předáváme jednotlivým aktivitám při jejich spouštění či volání. Jedná se vlastně o zavolání aktivity, kterou potřebujeme nějakým způsobem použít při běhu naší aplikace, či dostat její výstupní data. Spolu s voláním jiné aktivity jí předáváme i informace potřebné pro běh nové aktivity.

```
Intent novinky_play = new Intent (MainActivity.this,  
Novinky.class);  
MainActivity.this.startActivity(novinky_play);
```

Při spuštění aktivity Zcu_Info se použije nové layout rozložení, tento layout je však pouhá část layoutu, který se nám ukáže. Protože budeme rozšiřovat třídu ListActivity, zcela nám postačí definovat jeden její řádek a všechny ostatní vytvoříme stejným způsobem, ale s jinými daty. Tato data vezmeme z res/raw/faculties (všechny soubory projektu se dají zavolat pouze podle jejich umístění v projektu - res/raw/faculties tedy znamená, že v adresáři našeho projektu je adresář res, v něm adresář raw a ten obsahuje soubor faculties.xml, v java kódu by se poté jednalo o volání R.raw.faculties), jedná se o XML soubor, kde atribut value obsahuje název jednotlivých fakult. Protože nepředpokládám časté změny v názvu fakult na Západočeské univerzitě, nebo kolejí, je toto stálý soubor obsažený přímo v instalačním balíčku apk. Soubory obsažené v úložišti resources nejde v aplikaci nijak měnit.

Po otevření souboru a vytvoření instance třídy InputStream tohoto souboru použijeme zabudovanou logiku pro zpracování XML kódu a tím vytvoříme dokument typu DOM. Poté vybereme elementy jednotlivých slov a uložíme atributy do adaptéru typu ArrayAdapter. Tento typ adaptéru již může zpracovávat widget typu ArrayList, který nám i vytvoří výčet jednotlivých fakult. Více s tímto souborem nebudeme pracovat, tudíž jej hned po načtení potřebných dat zavřeme.

Pro práci s výčtem napřed vyzkoušíme, zda se dostaneme k námi požadovanému souboru. Pokud by tomu tak nebylo, nebo soubor faculties.xml neexistoval, objevila by se nám toast s chybovou hláškou (toastům a jejich vyvolání je věnována část níže).

Po provedení výběru fakulty, nebo koleje, pomocí callback metody `onItemClickListener()` zjistíme pozici vybraného prvku. Podle toho zobrazíme příslušné informace o studijním oddělení příslušné fakulty (nebo správci koleje), včetně úředních hodin. Tyto informace byly získány z webových stránek jednotlivých fakult.

Během zobrazení informací o studijním oddělení nám aktivita dává možnost napsat na studijní oddělení e-mail, nebo zavolat přímo do kanceláře.

V následujícím příkladu je použito přiřazení potvrzovacího tlačítka do kódu Java, abychom s ním mohli interagovat. Po kliknutí, přejdeme do aktivity Dialer.

```
Button dial=(Button) findViewById(R.id.dial);
dial.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        String toDial = "tel:"+
number.getText().toString();
                                startActivity(new
Intent(Intent.ACTION_DIAL,
                                Uri.parse(toDial)));
    }
});
```

Toast

Android nám nabízí několik možností, jak můžeme uživatele upozornit na to, že aplikace právě něco dělá, nebo již dodělala a jaký je výsledek. Jednou z nich je zobrazení toast. Toast nijak nedeaktivuje právě běžící aplikaci. Jedná se o zprávu, která se sama zobrazí a také sama zmizí. Nemůžeme tedy jistě vědět, že si uživatel zprávu přečetl a tudíž se nesmí jednat o zprávy zásadní. Tyto zprávy slouží hlavně k zobrazování ukončení procesů běžících na pozadí, nebo stavu baterie v nekritickém stavu.

Třída Toast má statickou metodu `makeText()`, která jako parametr přijímá instanci typu String (nebo ID textového řetězce) a vrací instanci typu Toast. Metoda `makeText()` ještě potřebuje instanci třídy Activity (nebo jiný kontext) a časový údaj, jak dlouho bude toast viditelný. Pro životnost toastu se užívají konstanty `LENGTH_SHORT` a `LENGTH_LONG`. Toast vyvoláme jednoduchým příkazem.

```
Toast.makeText(this, "Ahoj!", Toast.LENGTH_SHORT).show();
```

Kromě toastu ještě můžeme vyvolat Varování a Upozornění.

6.2 Aktivita Maps

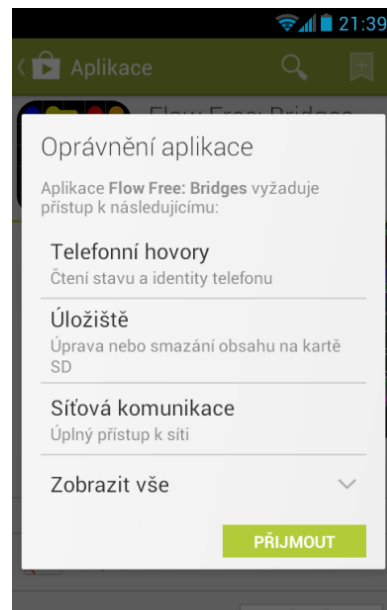
Aktivita Maps je jediná, která hned při spuštění v základním režimu potřebuje připojení k internetu a přístup k několika sensorům v hardwaru zařízení. Abychom však tento přístup od zařízení měli povolen, potřebujeme souhlas uživatele. Ten se provádí při instalaci aplikace do zařízení (obrázek 6-4). Aby však po nás systém požadoval přijetí, musíme o ně zažádat v Manifestu.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Tato povolení nám umožní navázat s aplikací přístup k internetu a určit naši polohu. Kdybychom použili samotné povolení ACCESS_COARSE_LOCATION byla by přesnost naší polohy velmi malá. Použitím ACCESS_FINE_LOCATION zvyšujeme požadavek na přesnost v určení naší polohy a tím získáváme přesnější informace o naší poloze.

Manifest je XML soubor, který nám pomáhá dávat jednotlivé aktivity dohromady, jednodušeji používat služby a celkově vypovídá o tom, co všechno naše aplikace bude umět a jak se bude projevovat vzhledem k systému Android, tak i k ostatním aplikacím. Například prvky intent-filter a action můžeme definovat hlavní aktivitu naší aplikace a pomocí prvku category definujeme, že tato aktivita se spustí po kliknutí na její ikonu v menu.



Obrázek 6-4 Potvrzení oprávnění v obchodě Google Play

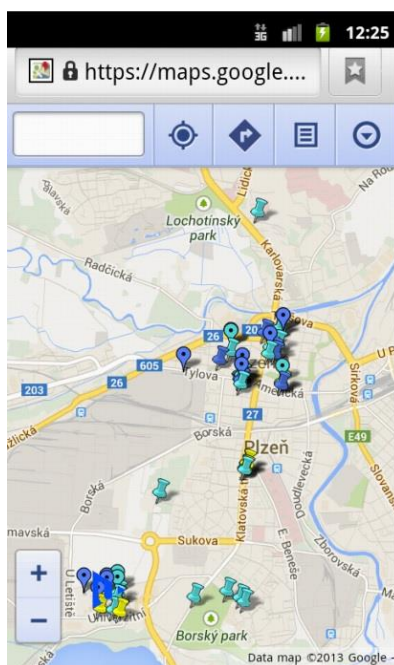
```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER"
/> </intent-filter>
```

Pro vytvoření naší mapy a práci s ní musíme vytvořit podtřídu třídy MapActivity. Jedná se o podobný model jako v případě ListActivity, kdy jsme obalili část logiky této aktivity. Ve třídě MapActivity dominuje widget MapView. Protože widget MapView není součástí jmenného prostoru com.google.android.widget, musíme do deklarace tohoto prvku přidat element com.google.android.maps.MapView. Po této úpravě již widget funguje bez problémů. Musíme však ještě doplnit dva atributy, které po nás widget bude vyžadovat. Jedná se o:

- android:apiKey – což je odkaz na API klíč služby Google Maps.
- android:clickable – pokud chceme uživateli povolit interakci s mapou, nastavíme na hodnotu true, pokud nechceme, aby uživatel mohl s mapou pohybovat, a případně vyvolávat interakce dotykem nastavíme na false

Abychom mohli tuto mapu používat, musíme opět upravit soubor Manifest.xml přidáním elementu <uses-library> a dát mu atribut android:name = “com.google.android.maps“. Tímto pokynem dáme systému najevo, že budeme používat i volitelnou API Maps.

Pokud chceme uživateli umožnit větší kontrolu nad zobrazenou mapou, musíme ještě prostřednictvím metody getMapController() získat objekt typu MapController. Tento objekt nám poskytuje široké možnosti kontroly. V aplikaci jsou pouze použity přibližovací lišta a centrování mapy na aktuální polohu.



Obrázek 6-5 Zobrazení všech bodů zájmu v aplikaci

Přibližování ovládáme pomocí metody `setZoom()` objektu typu `MapController`. Tato metoda nám umožní nastavit přiblížení mapy v rozsahu 1 až 21, kde 1 znamená zobrazení celého světa. Doporučovaná hodnota se pohybuje mezi 14 a 17 podle toho, jak velkou plochu mapy chceme zobrazit.

Centrování je o něco složitější vzhledem k tomu, že metoda `setCenter()` přijímá jako parametr objekt typu `GeoPoint`. Objekt `GeoPoint` vyjadřuje zeměpisnou polohu pomocí zeměpisné délky a šířky jako integery skutečné délky a šířky vynásobené exponentem $1E6$. Jedná se především o šetření místa v paměti a o rychlost některých interních výpočtů.

```
map= (MapView) findViewById(R.id.map);

map.getController().setCenter(getPoint(40.76793169992044, -
73.98180484771729));

map.getController().setZoom(17);
map.setBuiltInZoomControls(true);
```

6.2.1 Vrstvy

Google Maps nám nabízí možnost skutečnou mapu překrýt novou vrstvou. Tato vrstva může obsahovat například vyznačené body zájmu. Vrstev může být na sobě více. V každé vrstvě vytvoříme jeden z bodů zájmu, ty poté budeme moci vyhledávat podle čísla vrstvy. Tato skutečnost nám značně usnadní vyhledávání bodů zájmu.

Na přidání vrstvy použijeme metodu `Add()` třídy `Overlay`. Viz příklad níže.

```
Drawable marker =
getResources().getDrawable(R.drawable.marker);

marker.setBounds(0, 0, marker.getIntrinsicWidth(),
marker.getIntrinsicHeight());

map.getOverlays().add(new SitesOverlay(marker));

me=new MyLocationOverlay(this, map);
map.getOverlays().add(me);
```

V tomto příkladu přidáme do mapy body vyznačené objektem `marker`, který symbolizuje umístění těchto objektů, a poté přidáme naši aktuální polohu v nové vrstvě.

Na získání naší aktuální polohy používáme metodu `getLastLocation()`. Tato metoda spolu s aktualizací naší polohy na mapě po časovém úseku (nebo změny polohy o více než 10 metrů) ukazuje, kde právě jsme a zda není v našem okolí některý z modů zájmu, který by byl zobrazen v jiné vrstvě.

6.2.2 Offline mapy

Pro uživatele, kteří nebudou chtít (nebo moci) se připojit k internetu, nabízí aplikace zobrazení JPG souboru s mapou Plzně s vyznačenými budovami ZČU. Tato mapa byla převzata z Průvodce prváka [8]. Soubor je zobrazen v Layoutu WebView, ten nám nabízí možnosti webového prohlížeče, jako je například přiblížení, nebo oddálení obrázku.



Obrázek 6-6 Zobrazení offline mapy v landscape režimu pro lepší čitelnost na zařízení

6.3 Aktivita Menza

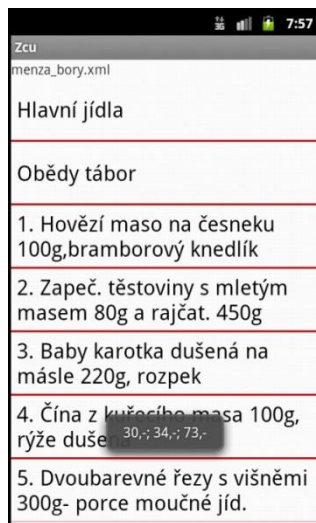
Tato aktivita pracuje s XML soubory, ve kterých jsou uloženy jídelníčky obou plzeňských menz. Dle preferencí nastavených uživatelem se může zobrazovat pouze bližší, či pouze jedna menza. Možnost výběru preferencí je možný jak v hlavní aktivitě po stisknutí tlačítka Menu, tak v aktivitě Preferences. Po spuštění aktivy je možné zobrazit jídelníček pouze jediné menzy. Pokud uživatel žádnou menzu nevybral, zobrazí se mu jídelníček menzy na borech v Univerzitní ulici. Jaká menza je vybraná, se zobrazí uživateli v horní části displeje.

Výběrem některého z jídel ve výčtu vytvoří Toast, který uživateli sdělí cenu pro studenta ZČU, zaměstnance ZČU a externí osoby v tomto pořadí.

Oba soubory (jeden pro každou menzu) jsou uloženy na interním úložišti a při ukládání jsou jim dány takové hodnoty, aby je nemohla vidět žádná jiná aplikace v zařízení (viz. Aktivita Aktualizace).

Struktura souboru menza_bory.xml je následující:

```
<resources>
<akce value = "Borůvkový koláč s drobenkou 320g" price = "36,-
; 41,- ; 78"/>
</resources>
```



Obrázek 6-7 Zobrazení Toastu po výběru jedné z položek ve výčtu

6.4 Aktivita Novinky

Aktivita Novinky pracuje s předem zpracovaným a upraveným seznamem aktualit z webové adresy <http://www.zcu.cz>. O toto zpracování se stará aktivita Aktualizace, která zároveň celý seznam uloží jako XML soubor na interní úložiště, a je popsáno v kapitole 6.9. Ve výčtu zobrazuje novinky od nejaktuálnější po tu nejméně aktuální. Každá položka výčtu je zároveň vybavena odkazem, jehož adresa se rozbálí po kliknutí na příslušnou položku výčtu. Výčet je několikařádkový, takže ani delší aktuality nemají problém se zobrazením. Pokud je ve výčtu více položek, než je možné zobrazit na jednu obrazovku, vytvoří se posuvný výčet.

Formát XML souboru je velmi podobný jako u souboru `menza_bory.xml`

Aktivita Novinky pracuje s XML parserem, který postupně projde námi vybraný soubor a uloží jeho atributy `value` do připraveného layoutu pro výčty. Zároveň uloží jeho druhý atribut `url` do paměti, abychom jej v případě výběru uživatele mohli hned zobrazit pomocí layoutu `WebView`. Tímto layoutem (`WebView`) se nám na základě URL adresy zobrazí pouze daná aktualita. V příkladu je vidět stejný postup jako u aktivity `Zcu_Info`. Protože tento postup používáme (díky zobrazování informací pouze ze souborů) ve většině případů v této aplikaci, vytvořil jsem metodu `fillInList()`.

```
void fillInList(String sourceFile, String elementTagName,  
String att1, String att2, Context context)
```

- `SourceFile` reprezentuje jméno zdrojového XML souboru
- `elementTagName` určuje, které elementy v souboru hledáme
- `att1` určuje první atribut, jehož hodnotu budeme zobrazovat ve výčtu
- `att2` určuje druhý atribut, hodnota tohoto atributu je URL adresa příslušné Novinky ve výčtu
- `context` současný stav zařízení

Tato metoda nejdříve načte soubor s příslušným jménem jako jednu proměnnou typu `String`. Tu poté upraví tak, aby ji dokázal zpracovat DOM parser. Ten nám zároveň uloží jednotlivé Novinky do jednoho `ArrayListu` a jejich `url` adresy do druhého `ArrayListu`.

6.5 Aktivita Udalosti

Aktivita Udalosti pracuje na stejném principu jako předchozí aktivita (Novinky), rozdílná je pouze tím, že informace o událostech bere z URL adresy <http://www.portal.zcu.cz> a zobrazuje jako první část zprávy datum, kdy se tato akce bude konat. Struktura XML souboru je obdobná jako u Novinek a je použita i stejná metoda pro získávání dat z tohoto souboru. Rozdílem je poté možnost, po vybrání události z výčtu, na tuto akci poslat pozvánku prostřednictvím e-mailu.

Tato možnost se zobrazí pomocí AlertDialogu. V první části kódu vidíme způsob vyvolání tohoto dialogu a nastavení titulku a zprávy v dialogu. Momentálně se jedná o název vybrané akce.

```
new AlertDialog.Builder(this)
    .setTitle("Podrobnosti o události")
    .setMessage(items.get(position).toString())
```

V druhé části vyvoláváme neutrální tlačítko, které má implementováno vlastní posluchač. Díky tomu zjistíme, zda bylo tlačítko stisknuto. Po jeho stisknutí vytvoříme nový záměr s požadavkem na odeslání e-mailové zprávy. Pomocí záměru také předáme volané aktivitě také informaci o předmětu e-mailu a o jeho textu, do kterého nastavíme url adresu události. Poté aktivitu spustíme.

```
.setNeutralButton("Pozvat", new
DialogInterface.OnClickListener() {

public void onClick(DialogInterface dlg, int sumthin) {

Intent emailIntent = new Intent(Intent.ACTION_SENDTO,
Uri.fromParts("mailto", "", null));
emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Pozvanka na
akci");
emailIntent.putExtra(Intent.EXTRA_TEXT,
urls.get(position).toString());
startActivity(Intent.createChooser(emailIntent, "Send
email..."));
}).show();
```

6.6 Preference uživatele

Jedním ze zadání práce je, aby měl uživatel možnost zvolit vlastní preference. Možnosti, které si uživatel pomocí této aktivity může nastavit, jsou: připojení k internetu (aplikace sama připojí zařízení k internetu, pokud bude internetové připojení k dispozici a při spuštění zaktualizuje všechny soubory), změna barev aplikace (pouze na 3 předem navržené možnosti), výběr menzy a nastavení uvítací zprávy. Z možných preferencí jsem vybral jako názorný příklad možnost zobrazovat uvítací zprávy dle výběru uživatele.

S tvorbou preferencí nám opět pomohou XML soubory, které (podobně jako layouts) dokáží specifikovat vizuální rozhraní. Následující XML kód specifikuje naše nastavení preferencí.

```
<PreferenceScreen
xmlns:android="http://schemas.android.com/apk/res/android" >

  <PreferenceCategory
    android:key="category"
    android:title="Ostatni" >
    <CheckBoxPreference
      android:defaultValue="true"
      android:key="perform_welcome"
      android:summary="Aplikace bude zobrazovat uvitaci zpravu"
      android:title="Zobrazovat uvitaci zpravu" />

    <EditTextPreference
      android:defaultValue="Ahoj!"
      android:dialogMessage="Napiste zpravu"
      android:dialogTitle="Uvitaci zprava"
      android:dependency="perform_welcome"
      android:key="welcome_message"
      android:summary="Vytvorte si vlastni uvitaci zpravu"
      android:title="Uvitaci zprava" />
    </PreferenceCategory>
</PreferenceScreen>
```

V preferencích máme jeden widget typu CheckBox. Pokud uživatel změní hodnotu tohoto widgetu na true, tedy zaškrtně widget, bude naše aplikace vědět, že při každém spuštění má zobrazovat uvítací zprávu. Druhý widget typu EditText nám dává více svobody. Díky němu po rozkliknutí můžeme napsat svojí vlastní uvítací zprávu. Zpráva může být v jakémkoliv tvaru. Druhý widget je díky vlastnosti

`android:dependency="perform_welcome"` přímo spojený s prvním widgetem a pokud není CheckBox zaškrtnut, možnost vybrat uvítací zprávu se nám ani nezobrazí.

Aby zvolené preference byly použity v celé aplikaci a nejenom v aktivitě Preferences, musíme použít v .java kódu metodu `getDefaultSharedPreferences()`. Ta nám zajistí, že preference budou distribuovány po celé aplikaci.

```
SharedPreferences prefs =  
PreferenceManager.getDefaultSharedPreferences(this);
```

6.7 Aktivita Pomoc

Tuto aktivitu jsem zařadil po vzoru aplikace Harvard University. Jedná se o jednoduchý seznam s čísly na hasiče, policii a rychlou záchrannou službu, stejně tak na integrovaný záchranný systém. Kromě situací, kdy bude uživatel potřebovat jednu ze složek záchranného systému, může nastat i moment, kdy bude potřebovat kontaktovat HelpDesk CIV, či něco zapomene v prostoru univerzity. Proto jsou přidány i tlačítka pro vytočení HelpDesku, nebo napsání mailu. Po kliknutí na některé z čísel se objeví aktivita DIAL, která bude čekat na potvrzení odchozího hovoru. Pro případ, že se uživatel pouze překlíkl. V České republice taky existují tři výstražné signály, které jsou sdělovány veřejnosti pomocí sirén. Tato aplikace popisuje všechny tři signály a pokyny pro případ, že je uživatel uslyší.

6.8 Full Web

Zobrazení webové stránky <http://www.zcu.cz> v uživateli preferované aplikaci internetového prohlížeče. Zvoleno především pokud by uživatel vyhledával další informace, které v aplikaci nejsou. Volání metody je provedené pomocí Intentu.

```
String url = "http://www.zcu.cz";  
Intent i = new Intent(Intent.ACTION_VIEW);  
i.setData(Uri.parse(url));  
startActivity(i);
```

6.9 Aktivita Aktualizace

Aktivita starající se o aktuálnost souborů v aplikaci. Pro její správné fungování musíme do manifestu přidat následující část kódu:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

První řádek slouží k připojení k internetu, druhý k přístupu k funkci, která nám řekne, zda máme přístup k síti internet. Pokud bychom přístup neměli a okamžitě se snažili stahovat data, byla by nahlášena chyba a naše aplikace by mohla být předčasně ukončena.

Příklad kontroly připojení k internetu použitý v aktivitě MainActivity:

```
private boolean isNetworkConnected() {
    ConnectivityManager cm = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo ni = cm.getActiveNetworkInfo();
    if (ni == null) {
        // Neexistuje připojení k internetu
        return false;
    } else
        return true;
}
```

Pro připojení a parsování HTML stránek používám třídu `remoteContent()`, která pomocí externí knihovny Jsoup-1.7.2.jar (která je dodaná v instalačním apk balíčku), parsuje data jednotlivých webových stránek a ukládá do `HashMap<String, String>` námi zvolená data. Jedná se především o názvy aktualit, nové události a jejich url adresy. Externí knihovna Jsoup-1.7.2.jar byla, z důvodu její velikosti v originální podobě, zbavena nepotřebných tříd a metod a její velikost je tak zmenšena o více než polovinu.

Parsování webových stránek

O parsování webových stránek se stará externí knihovna Jsoup-1.7.2. Tato knihovna načte celou html strukturu stránky do objektu Dokument. V tomto dokumentu provedeme hledání přesné cesty dle tagů hledaného elementu za pomoci nám známé struktury html souboru. Poté vytvoříme `HashMap<String, String>` kterou zaplníme hodnotami elementu, nebo atributu v určeném elementu. Parser ukládá najednou hned několik hodnot, které jsou pro naši aplikaci důležité. Například u novinek se tak jedná o jméno novinky a url adresu pro zobrazení více podrobností. U menzy by se jednalo o název jídla a její ceny. Musíme si dávat pozor při ukládání url adresy a to z důvodu, že v html kódu není uvedeno jméno serveru, ale pouze pokračování adresy, na které se právě nacházíme. V příkladu vidíme metodu `fetchZcuNovinky()`.

```
Document doc;

doc = Jsoup.connect(URL_ZCU).get();

Elements lis = doc.select("div[id=roz_aktuality] >
ul[class=content] > li");
int i=0;
for (Element li : lis) {

    i++;

    HashMap<String, String> output = new HashMap<String,
String>();
    Elements spans = li.getElementsByTag("span");

    Elements anchors = li.getElementsByTag("a");

    if (i==1){
        output.put("odkaz",URL_ZCU + anchors.get(0).attr("href"));
        output.put("text", anchors.get(0).text());
        output.put("datum", "datum v textu");
        list.add(output);
        continue;
    }

    output.put("odkaz", URL_ZCU + anchors.get(0).attr("href"));
    output.put("text", anchors.get(0).text());
    output.put("datum", spans.get(0).text());
    list.add(output);
}
```

V aplikaci ale pracujeme s hodnotami, které vyjadřují titulek a webový odkaz (pro aktivitu Novinky). Uživateli je zobrazen pouze titulek v prostředí výčtu a po vybrání některého z titulků je odkázán na webovou stránku tohoto článku, která se zobrazí v prostředí WebView bez možnosti navigace mezi dalšími stránkami.

Podobné parsery jsou v aplikaci celkem 4. Dva v aktivitě Menza – pro každou z menz jeden a po jednom také v aktivitách Novinky a Události. Liší pouze načítanou stránkou a umístěním hledaných elementů a atributů.

Ukládání dat do souborů

Všechna získaná data budeme okamžitě zapisovat do souborů, které jsou přítomny v interním úložišti zařízení. Abychom mohli tyto soubory aktualizovat, budeme je nejprve potřebovat otevřít a spolu s novými daty do nich zapsat námi potřebnou strukturu XML. Jinak by se nám data v jednotlivých aktivitách nezobrazila správně. Stará data jsou při každé aktualizaci přepsána díky nastavení `Context.MODE_PRIVATE`. Tento kontext zároveň zajistí, že žádná jiná aplikace nebude mít přístup k našim datům.

Otevření souboru v privátním módu.

```
FileOutputStream fos;

try{
fos = openFileOutput(filename,Context.MODE_PRIVATE);

XmlSerializer serializer = Xml.newSerializer();
```

Vložení všech potřebných náležitostí do XML souboru. První tag našeho souboru bude `<resources>`, poté budeme přidávat tagy `<akce>` s prvním atributem `value` do kterého ukládáme název novinky (nebo události). Druhý atribut měníme v závislosti na cílovém souboru, proto je nastavený jako proměnná. V momentě, kdy máme všechny elementy uloženy, přidáme koncový tag a dokument ukončíme.


```
serializer.setOutput(fos, "UTF-8");
serializer.startDocument(null, Boolean.valueOf(true));
serializer.startTag(null, "resources");

for (int i = 0; i < infos.size(); i++) {
serializer.startTag(null, "akce");
serializer.attribute("", "value", infos.get(i).get("text"));
serializer.attribute("", attr2, infos.get(i).get("odkaz"));
serializer.endTag(null, "akce");
}

serializer.endDocument();
serializer.flush();
fos.close();
```

6.10 Menu

V aplikaci je také implementováno menu pro případ, že by uživatel stiskl hardwarové tlačítko Menu. Toto menu je značně zjednodušené, protože se v něm nacházejí pouze tři volby. Zobrazení informací o aplikaci, celkové ukončení aplikace a možnost zvolení menzy.

Menu je specifické tím, že při vyvolání vždy zobrazí pouze prvních 5 prvků (v případě že máme více prvků) a další až po rozbalení celého menu. Prvkům můžeme měnit jednotlivé priority zobrazení a tím zajistit, že zobrazené prvky budou ty nejpoužívanější, nebo pro uživatele nejužitečnější. Zároveň každý z prvků menu může fungovat k vyvolání submenu. To má stejné vlastnosti jako aktuální menu pouze s tím rozdílem, že jeho prvky už nemohou vyvolat další submenu.

Kromě pořadí prvků můžeme specifikovat i další vlastnosti menu, jedná se především o:

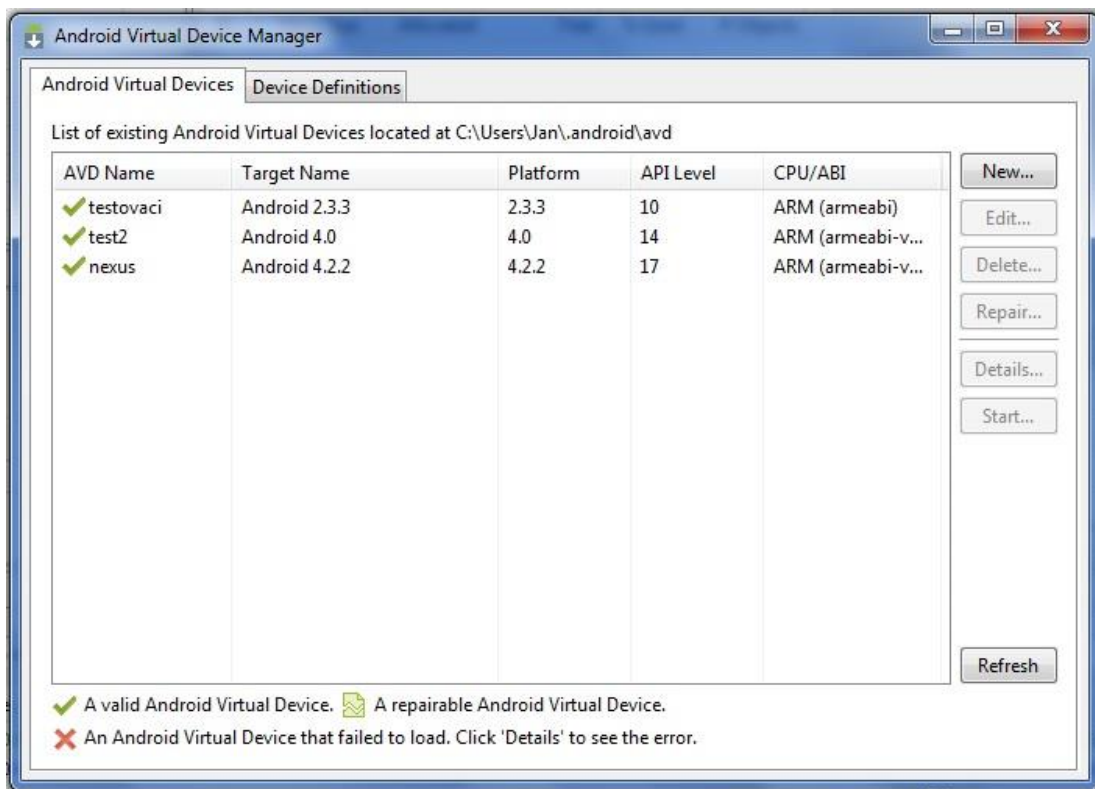
- **Text a obrázek** – výčet menu nemusí být jenom textový, ale může být použita i ikona nebo obrázek. Tyto prvky jdou i kombinovat a pomocí layoutů můžeme vytvořit i velmi komplikované prvky výčtu.
- **Aktivace** – Zda je prvek v menu aktivní, nebo ne, může být určeno jak přímo v návrhu, tak můžeme tento atribut dynamicky měnit v závislosti na dalším výběru uživatele. I neaktivní tlačítko je viditelné.

- **Viditelnost** – Stejně jako u aktivace i u viditelnosti lze tento stav dynamicky měnit v závislosti na ostatních interakcích uživatele. Neviditelné tlačítko je také deaktivované.
- **Klávesová zkratka** – Každý prvek v menu může mít svoji numerickou, nebo písemnou zkratku. Jedná se o jedno číslo, nebo jedno písmeno, jehož stiskem označíme prvek. Tato vlastnost se vytrácí společně s hardwarovými klávesnicemi, protože vyvolávat softwarovou klávesnici společně s menu je pro běžného uživatele neefektivní.

7 Testování aplikace

Testování aplikace probíhalo na telefonu HUAWEI Y300-0100 s Androidem 4.2.2 (API Level 17), telefonu Nexus S s Androidem 4.1.2 (API Level 16) a dvou AVD (Android Virtual Device) zařízeních s rozdílnými konfiguracemi. První konfigurace nazvaná „testovací“ byla vybavena API Level 10, 3,3 palcovou obrazovkou 240 x 400 a RAM 512 MiB.

Druhé testovací virtuální zařízení bylo stejné konfigurace jako telefon Google Nexus 7. Tzn. API level 17, 7,3 palcová obrazovka 800 x 1280 a paměť RAM o velikosti 1024 MiB.



Obrázek 7-1 Android Virtual Device Manager

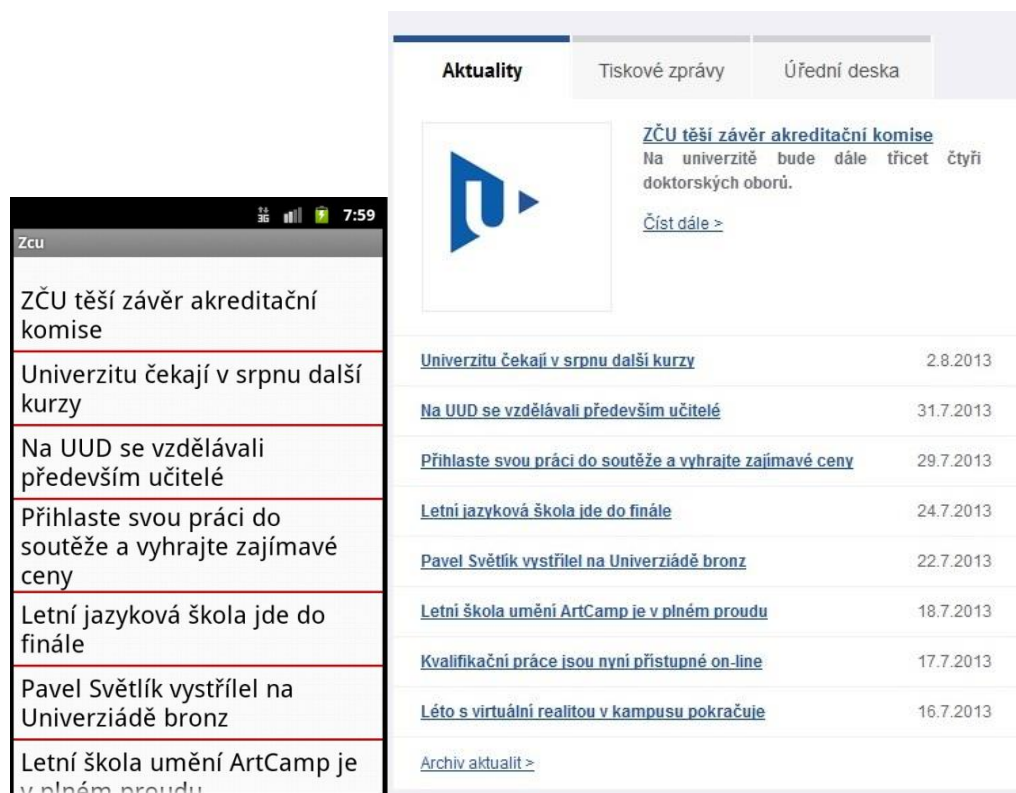
7.1 Běh aplikace

Běh aplikace je velice plynulý. I s přístupem k internetu nejsou odmlky delší než čas potřebný k načtení webové stránky, nebo k zavolání potřebné služby. Při aktualizaci všech dat najednou je odmlka kolem tří vteřin. Za tuto dobu se načtou 4 webové stránky, nalezne se v nich potřebná informace a ta se uloží do 4 nezávislých souborů. Při postupné aktualizaci je prodleva kolem jedné sekundy. Aplikace však nepotřebuje stahovat data častěji než jednou denně a proto by tento problém neměl mít vliv na další fungování aplikace.

Testována byla především správnost stažených dat, určení správné polohy a jednání na základě zvolených preferencí.

7.2 Správnost Dat

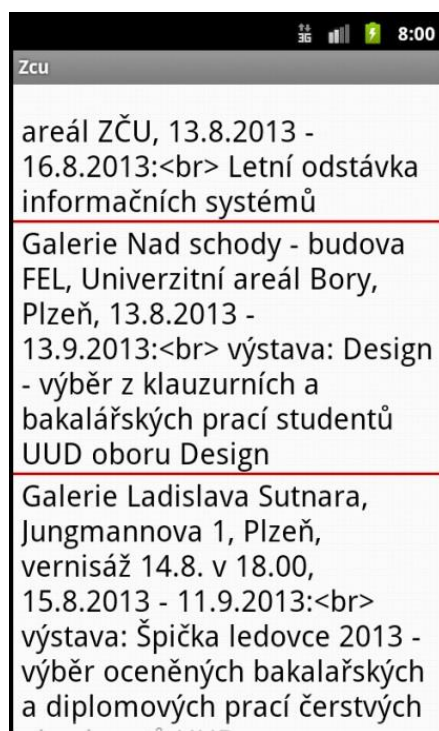
Správnost stáhnutých dat byla několikrát ověřena dle původního poskytovatele obsahu. Korektnost byla ověřena u všech poskytovatelů.



Obrázek 7-2 Aktuality zobrazené v aplikaci a na zcu.cz



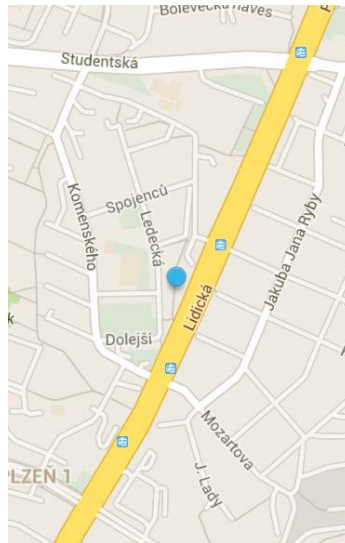
Obrázek 7-3 Zobrazení událostí na portal.zcu.cz



Obrázek 7-4 Zobrazení Událostí v aplikaci

7.3 Určování polohy

Dalvik Debug Monitor Service (DDMS) nám poskytuje možnost zasílat virtuálnímu zařízení informace o naší aktuální poloze. Pomocí této služby jsem vyzkoušel několik pozic na území Plzně. Aplikace dokázala simulovanou polohu bez problémů určit s naprostou přesností. Na reálném zařízení je přesnost menší z důvodu, že informace o poloze jsou zkresleny odchylkou v měření senzorů.



Obrázek 7-5 Určení polohy na virtuálním zařízení

7.4 Jednání dle preferencí

Jako základní nastavení jsou oba CheckBoxy (připojení k internetu a uvítací zpráva) neaktivní, barva zvolena bílá a preferovaná menza zvolena v Univerzitní ulici v areálu univerzity. Preference fungovali bez problémů.

8 Možnosti rozšíření aplikace

Součástí práce je i analýza potenciálních možností a rozšíření aplikace o nové funkce.

V závislosti na testování je první věcí, která by šla na aplikaci změnit práce s daty. Použitím nepřímého přístupu bychom ušetřili jak výkon zařízení, tak i místo na úložišti. Další z úprav aplikace by mohla být větší kooperace s požadavky uživatele a přidání několika dalších zdrojů, které by pro uživatele byly důležité. Také některé layouty a zobrazení by šly dále optimalizovat. Přidání obsluhy rotace zařízení.

Možnosti jednotlivých aktivit v aplikaci jsem postupně rozepsal podle jména aktivity.

Zcu_Info

Nezobrazovat pouze informace o studijním oddělení, ale o celé fakultě, možnost přihlášky a prohlížení jednotlivých kurzů, které na dané fakultě fungují.

Maps

Upravit v zobrazování více potencionálních bodů zájmu. Přidat možnost vložení vlastního bodu, či jeho smazání. K bodům zájmu přidat otvírací doby.

Menza

Změnit na Správu kolejí a menz a přidat i možnost výběru koleje a bufetů.

News

Přidat více informačních zdrojů, například v souladu s preferencemi uživatele. Informační zdroje nebrat pouze ze serverů ZČU, ale i z důvěryhodných serverů města Plzně.

Events

Přidat více informačních zdrojů, například v souladu s preferencemi uživatele. Možnost ukládat události do interního kalendáře.

Nové funkce

Možnost přihlašování na Portál (portal.zcu.cz), hledání v on-line katalogu knihovny, off-line návod na připojení mobilních zařízení spolu s certifikátem. Možnost naladění rádia Bomba.

Webové služby ZČU

Aplikace v současné době nevyužívá žádnou z webových služeb poskytovaných ZČU. Mezi jedny z prvních nových aktivit by bylo vhodné zařadit práci na Portále a zároveň i využívání webových služeb poskytovaných ZČU.

Možnost ukládání dat

V aplikaci jsem řešil otázku jak ukládat data, která uživatel bude potřebovat pro interakci s aplikací. Zvolil jsem XML soubory. Především z důvodu existence množství nástrojů pro práci s těmito soubory. Pro uložení a zobrazování některých dat by mohla být lepší SQLite databáze, ale data nejsou v žádném z implementovaných případů natolik velká. Celková velikost aplikace je po aktualizaci zdrojů 3,33 MB z čehož 2,3 MB tvoří plán Plzně pro zobrazení v případě, že aplikace není připojená k internetu.

9 Závěr

V rámci práce jsem se seznámil s nejrozšířenějším operačním systémem pro mobilní zařízení a vytvořil aplikaci pro studenty ZČU. Také jsem prošel množstvím nabízených senzorů v mobilních zařízeních, které se stále zvětšuje a implementoval všechny, pro tuto aplikaci, vhodné. Aplikace pracuje i s daty, které jsem získal prostřednictvím těchto senzorů. Jedná se především o geolokační a polohové senzory.

Výsledkem práce je aplikace pro platformu Android, která studentovi předkládá nejenom informace o univerzitě, ale také přistupuje přímo k informačním zdrojům univerzity a uživateli je prezentuje v nenásilné formě. Zároveň ukazuje studentovi jeho aktuální polohu vzhledem k objektům ZČU a dokáže mu ukázat předem zvolenou místnost. Student si i na základě preferencí může určit, zda bude aplikace pracovat připojena k internetu, nebo bude pracovat v režimu off-line.

Přístup k informačním zdrojům je řešen přímo aplikací. Není využíván žádný server, který by komunikoval se zdroji místo aplikace a data poté posílal do aplikace. Z toho důvodu bylo důležité ošetřit, aby komunikace mezi aplikací a zdrojem byla co nejrychlejší a nezatěžovala zařízení pro další fungování zařízení.

Aplikace má do budoucna velký potenciál využití a představuje velkého pomocníka pro nové studenty ZČU. Ty dokáže seznámit s prostředím univerzity a pomůže jim vyznat se v rozsáhlém areálu ZČU v Plzni.

Použité termíny

- **API**, *Application Programming Interface*, rozhraní pro programování aplikací
- **Layout**, rozložení – používáno v souvislosti s rozložením obrazovky aplikace
- **XML**, *Extensible Markup Language* - obecný značkovací jazyk vyvinutý a standardizovaný konsorciem W3C (<http://www.w3c.org>)
- **Intent**, záměr – používáno v souvislosti s třídou Intent
- **HTTP**, *Hypertext Transfer Protocol* - internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML
- **SDK**, *Software Development Kit* - software umožňující vývoj pro určitou platformu

Literatura

[1] Android Developers [online]. 2013 [cit. 2013-07-13]. Dostupné z:
<http://http://developer.android.com/>

[2] Dashboards. Android Developers [online]. 2013 [cit. 2013-07-27]. Dostupné z:
<http://developer.android.com/about/dashboards/index.html>

[3] IDC: Analyze the Future [online]. ©2013 [cit. 2013-03-07]. Dostupné z:
<http://www.idc.com/>

[4] BlackBerry Developer [online]. ©2013 [cit. 2013-03-07]. Dostupné z:
<http://developer.blackberry.com/>

[5] Apple Developer [online]. ©2013 [cit. 2013-03-07]. Dostupné z:
<https://developer.apple.com/>

[6] Windows Phone Dev Center [online]. ©2013 [cit. 2013-03-07]. Dostupné z:
<http://developer.windowsphone.com/en-us>

[7] Android Development Tools, Marketplace Eclipse [online]. 2013
[cit. 2013-05-22]. Dostupné z: http://marketplace.eclipse.org/content/android-development-tools-eclipse#.UgKQQD_7DIM

[8] IAESTE PŘI ZČU PLZEŇ. Průvodce prváka 2012 [online]. Plzeň, 2012
[cit. 2013-07-14]. Dostupné z: http://www.iaeste.cz/content/2012/LC_Plzen/pp12.pdf

[9] Free Web Design Icons set.1 – PNG & PSD available. Vector Fantasy [online]. 2011 [cit. 2013-08-01]. Dostupné z: <http://www.vectorfantasy.com/icons-2/web-design-icons-png-psd-available/>

[10] MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací. Brno: ComputerPress, 2011, 375 s. ISBN 978-80-251-3194-7.

[11] MEIER, Reto. Professional Android 4 application development. Indianapolis: John Wiley, 2012, 817 s. ISBN 978-111-8262-153.

[12] HEROUT, Pavel. Java a XML. 1. vyd. České Budějovice: Kopp, 2007, 313 s. ISBN 978-80-7232-307-4.

[13] Position Sensors, Android Developers [online]. 2013 [cit. 2013-06-08]. Dostupné z: http://developer.android.com/guide/topics/sensors/sensors_position.html

[14] Android 4.3, Jelly Bean, Android [online]. 2013 [cit. 2013-08-01]. Dostupné z: <http://www.android.com/about/jelly-bean/>

[15] Studio, Android Developers [online]. 2013 [cit. 2013-08-01]. Dostupné z: <http://developer.android.com/sdk/installing/studio.html>

A Přílohy

A 1 CD

K práci je přiloženo CD s následující strukturou:

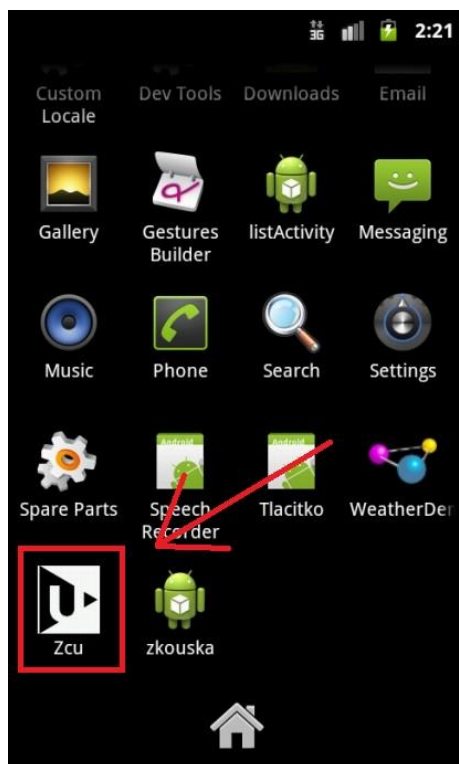
- /dokument/ - tento dokument v elektronické podobě
- /javadoc/ - vygenerovaný javadoc dokument
- /aplikace/ - zdrojové kódy aplikace

A 2 Uživatelská příručka

Průvodce základním používáním aplikace.

A 2.1 spuštění aplikace

Spuštění aplikace proběhne stisknutím příslušné ikony v základním menu Android. (viz. Obrázek A 2-1) Po načtení se zobrazí úvodní obrazovka.



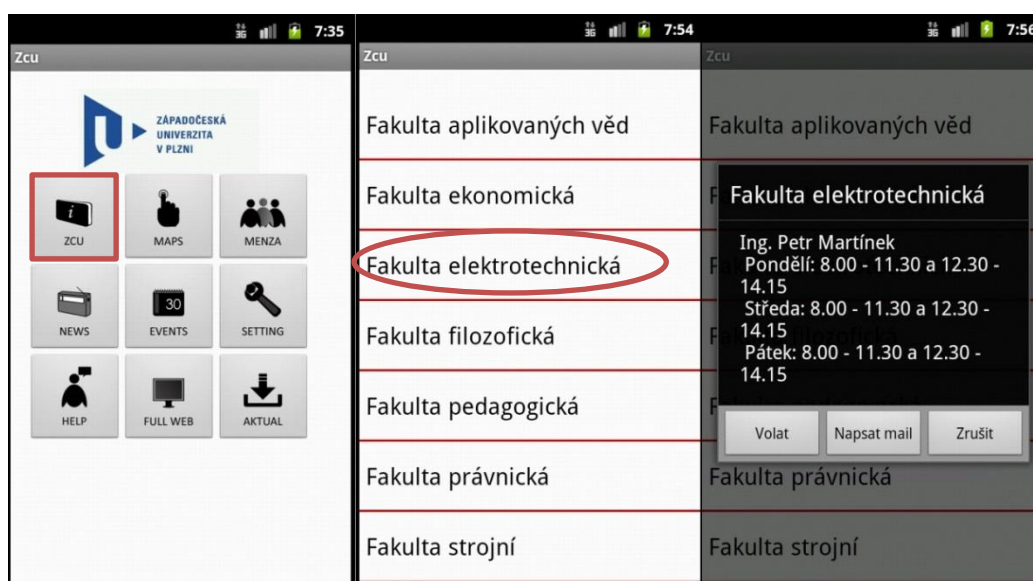
Obrázek A 2-1 Aplikace Zcu v Android menu

A 2.2 Ovládání aplikace

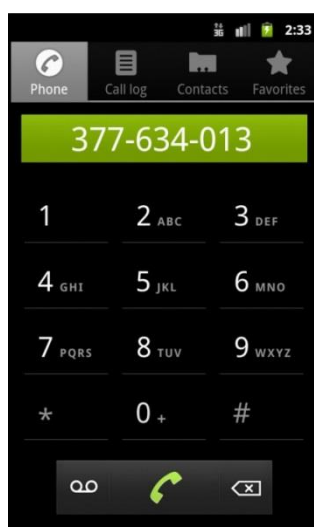
Ke spuštění vybrané aktivity slouží příslušná ikona. K jejímu spuštění na ni stačí kliknout. Pro návrat na předchozí obrazovku použijeme příslušné hardwarové tlačítko. Pro zobrazení Menu slouží hardwarové tlačítko. Aktualizaci dat můžeme provést v menu po stisknutí tlačítka AKTUAL.

ZCU INFO

Pro zobrazení informací o některém ze studijních oddělení klikneme v hlavním menu na ikonu ZCU. Zobrazí se nám výběr mezi všemi fakultami a všechny koleje. Poté dalším stiskem vybrané položky vybereme námi požadovanou položku. Zobrazí se nám okno s názvem našeho výběru, jménem pověřené osoby a úředními hodinami. Pokud nyní klikneme na tlačítko Volat, přejdeme na možnost zavolat dané osobě, jak vidíme na obrázku A 2-3. Pokud bychom chtěli vybrané osobě napsat e-mail, stiskneme tlačítko Napsat mail. Touto možností se dostaneme do námi preferovaného e-mailového klienta a již budeme mít nastaveného příjemce mailu a předmět. Předmět je nastaven na „Prosba o pomoc“, ale jde stále změnit.



Obrázek A 2-2 Postup při volbě studijního oddělení fakulty elektrotechnické



Obrázek A2-3 Pro uskutečnění hovoru musíme potvrdit zeleným sluchátkem

MAPS

Pokud jsme připojeni k internetu, zobrazí se nám po stisknutí tlačítka MAPS mapa se všemi objekty ZČU v Plzni a je jenom na nás, jestli si nějaký z nich vybereme. Nebo se pouze podíváme na rozložení budov univerzity. Pokud bychom připojeni k internetu nebyli, zobrazí se nám plánec, opět s vyznačenými budovami univerzity.

MENZA

Po stisknutí tlačítka MENZA se nám zobrazí aktuální jídelníček jedné z dvou menz v Plzni. Pokud bychom si vybrali některé z nabízených jídel a chtěli vědět, kolik stojí, stačí kliknout na jméno pokrmu a cena se nám sama zobrazí ve spodní části obrazovky. Zobrazí se tři hodnoty. První je cena jídla pro studenty, druhá pro zaměstnance a třetí pro externí osoby. Pokud si toto pořadí nepamätujeme, stačí kliknout na položku „Hlavní jídlo“ a zobrazí se nám malá nápověda. V horním levém rohu vidíme jakou máme právě zvolenou menzu.

NEWS

Po prvotní aktualizaci dat se nám po stisknutí tlačítka NEWS zobrazí přehled nejdůležitějších novinek. Pokud by Vás nějaká z nich zaujala a chcete se dozvědět více, stačí na ni kliknout a aplikace sama spustí webový prohlížeč s více informacemi.

EVENTS

Podobně jako u novinek, se zobrazí seznam nejbližších událostí. Akorát nyní po kliknutí na vybranou událost máme ještě možnost napsat mailovou pozvánku, ve které už je předem vložen odkaz na vybranou akci. Stačí doplnit doručovací adresu.

SETTING

Zde si můžeme nastavit menzu do které chodíme radši, nebo zobrazit informace o aplikaci.

HELP

Pokud se ve Vašem okolí právě něco stalo, tato služba Vám pomůže se zavoláním okamžité pomoci. Ale i v případě, že jenom potřebujete pomoct ohledně JISky, nebo Orion konta, můžete okamžitě zavolat na HelpDesk CIVu a pokud se Vám nechce volat, napište mail.

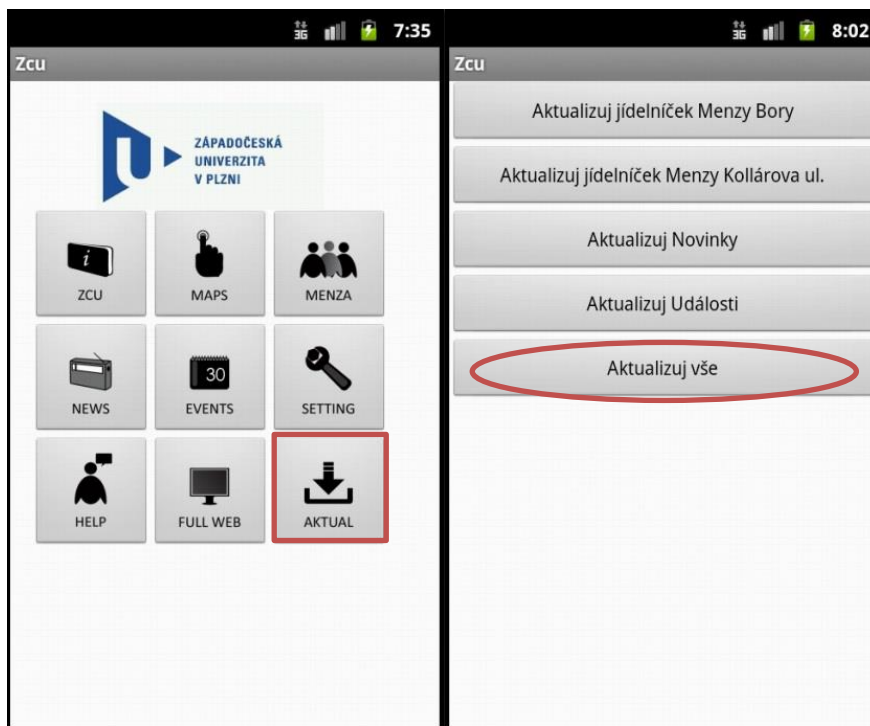
FULL WEB

Po zvolení této možnosti se Vám zobrazí v prohlížeči celá webová stránka Západočeské univerzity.

A 2.2 Aktualizace dat

Pro aplikaci je důležité, aby uživatel aktualizoval data v aplikaci uložená. Je to především z důvodu aktuálnosti jídelníčku v obou menzách, který se stahuje pouze na jeden den. Avšak i události a novinky se mění častěji než jednou týdně. Aby však uživatel nemusel pořád aktualizovat všechno, je zde možnost částečné aktualizace. V hlavním menu zvolíme ikonu AKTUAL. Tím se dostaneme do výběru, která data chceme aktualizovat. Můžeme také zvolit kompletní aktualizaci pomocí tlačítka „Aktualizuj vše“. Po prvním spuštění aplikace musíme aktualizovat data! Aplikace v tu chvíli vytvoří potřebné soubory a bude fungovat bez problémů. Kdybychom tak neudělali, hlásila by aplikace chybu s načítáním souborů.

Pokud dojde při aktualizaci dat k přerušení aplikace, budou data načtena bez ukončovacího znaku, a tudíž bude XML zdrojový soubor chybný. Při pokusu o načtení poškozeného souboru, bude zobrazena chybová hláška.



Obrázek A 2-4 Pro aktualizaci dat zvolíme tlačítko AKTUAL

A 3 Instalační příručka

Aby mohl uživatel nainstalovat aplikaci, musí v Nastavení > Zabezpečení – povolit položku „Neznámé zdroje“. Protože se aplikace nevyskytuje na Google Play, nešla by jinak na zařízení nainstalovat. Poté co si uživatel stáhne soubor Zcu.apk, ze stránky <http://home.zcu.cz/~blahaj/>, a dotykem ho otevře, začne automatická instalace. Ikona aplikace se objeví v Android menu. Nyní je aplikace připravena ke spuštění.