

UNIVERSITATEA ALEXANDRU IOAN CUZA IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Aplicație Android

Absolventă:

Alexandra-Mihaela Sirițeanu

Coordonator științific:

Lector Dr. Adrian Iftene

Sesiunea: iulie, 2012

Declarație privind originalitate și respectarea drepturilor de autor

Prin prezenta declar că Lucrarea de licență cu titlul “Aplicație Android” este scrisă de mine și nu a mai fost prezentată niciodată la altă facultate sau instituție de învățământ superior din țară sau străinătate.

De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sau indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imagini etc. preluate din proiecte *open-source* sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Declarație de consimțământ

Prin prezenta declar că Lucrarea de licență cu titlul “Aplicație Android”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Cuprins

Capitolul I.....	9
Introducere	9
Android	11
Platforma Hardware	11
Sistemul de operare	14
Componentele unei aplicații Android	15
Clasele de bază	15
Fișierul Manifest	16
Utilizarea de resurse externe.....	19
Rezumat	23
Capitolul 2	25
State of the Art.....	25
Aplicații de supravegheat persoane.....	25
Google Maps	25
Gypsi	26
LOCiMobile	27
Life360	28
Match2Blue	29
Glympse.....	29
Platforme de dezvoltare pentru Realitatea Augmentată.....	30
Look!	30
Unifeye SDK Mobile.....	31
Mixare	33

Wikitude	33
Layar.....	34
LibreGeoSocial.....	35
Rezumat	36
Capitolul III	37
Nivelul logic – Tehnologiile utilizate	37
Android Location Service	37
GPS Provider	38
Network Provider	39
Determinarea poziției unui telefon Android.....	40
LocationManager.....	41
Location Provider	42
Location.....	42
Criteria.....	42
LocationListener.....	42
Google Maps	43
Componentele Google Maps	43
JSONObject.....	44
Facebook API	46
Mixare API.....	48
Mesaje Text (Short Message Service).....	49
Nivelul de interfață.....	50
Lucrul cu animații	51
Utilizarea de liste cu conținut eterogen	53

Nivelul de date	54
Ghid de utilizare a aplicației	56
Rezumat	65
Concluziile lucrării.....	66
Bibliografie.....	68
Glosar	72

Tabel de figuri

Figura 1.1 - Arhitectura MCP	12
Figura 1.2 - Structura sistemului de operare Android.....	15
Figura 2.1 - Google Maps Latitude	26
Figura 2.2 - Gypsii.....	26
Figura 2.3 Aplicațiile LOCiMobile.....	27
Figura 2.4 - Life360.....	28
Figura 2.5 – Match2blue	29
Figura 2.6 - Glympse.....	30
Figura 2.7 - Structură Bibliotecă Unifeye	31
Figura 2.8 – Junaio	32
Figura 2.9 - Afișarea punctelor geografice utilizând Mixare	33
Figura 2.10 – Wikitude.....	33
Figura 2.11 – Layar	34
Figura 2.12 – LibreGeoSocial	34
Figura 3.1 - Sateliții GPS	36
Figura 3.2 Componentele Android pentru localizare.....	38
Figura 3.3 Înregistrarea aplicației MeetYou la Facebook.....	47
Figura 3.4 Ierarhie de obiecte View și ViewGroup.....	51
Figura 3.5 Translarea unui obiect de tip LinearLayou	53
Figura 3.6 Diagrama bazei de date	55
Figure 3.7 Crearea unui cont MeetYou și logarea în aplicați.....	57
Figure 3.8 Afișarea prietenilor pe hățile Google și pe ecranul telefonului utilizând Mixare.....	58

Figura 3.9 Crearea și vizualizarea unei locații utilizând Google Maps	59
Figura 3.10 Înregistrarea într-o locație	60
Figura 3.11 Recepționarea de cereri noi și afișarea lor	61
Figura 3.12 Editarea unui grup.....	62
Figura 3.13 Adăugarea unui utilizator a cărui cerere de prietenie a fost respinsă	62
Figura 3.14 Accesarea opțiunilor disponibile pe profilul unui prieten	63
Figura 3.15 Setarea cuvântului cheie	64
Figura 3.16 Căutarea unui prieten	64

Capitolul I

Introducere

Nevoia de socializare a omului a fost explorată încă din Antichitate, când Aristotel afirma: „Omul este prin natura sa o ființă socială” (Aristotel, Politica I), prin acest termen dorind să ilustreze faptul că entitatea umană este destinată traiului alături de semenii săi, în cadrul unei comunități organizate. Cum nu există societate fără comunicare, așa nu există niciun individ fără interacțiune în mediul social. De-a lungul timpului, formele de comunicare și perspectiva asupra acestui proces s-au dezvoltat în funcție de progresul tehnologiei. Apariția calculatorului și apoi a internetului a revoluționat modul de comunicare al oamenilor, reușind să depășească orice limitare dependentă de timp sau spațiu, ajutându-i să interschimbe într-un mod eficient informații. Totuși, niciunul dintre acestea nu a reușit să conecteze oamenii oriunde și oricum așa cum telefonul mobil o realizează. Apariția telefoanelor inteligente (smartphones) este o consecință a acestui fapt, după cum Benjamin Speckmann sugerează în lucrarea sa de disertație¹.

Necesitatea de a depăși orice prag în privința interrelaționării umane, precum și nevoia de a propaga în cel mai scurt timp informațiile sunt vizibile în vânzările telefoanelor mobile. Conform ziarului Gartner², vânzarea acestora a crescut cu 16.5% în trimestrul doi al anului 2011 față de același trimestru din anul precedent, ajungând la 428.7 unități. Vânzarea telefoanelor smart a crescut și ea cu 74%, fiind vândute 107.7 milioane de articole în întreaga lume. Android a fost sistemul de operare smartphone de top, cu o cotă de piață de 43.4% și cu o unitate de vânzare în valoare de 46,78 milioane dispozitive, depășindu-i pe Symbian aflat pe a doua poziție cu 22.1 % și Apple iOS cu 18.2%.

Dorința de a folosi telefonul nu numai pentru a apela pe cineva, pentru a expedia sau recepționa un mesaj a fost preconizată de Google încă din 2005, motiv pentru care a și cumpărat dezvoltatorul inițial de software pentru telefoane mobile Android Inc. în luna august a aceluiași an.

¹ Speckmann, B. 2008. The Android mobile platform. A Review Paper Submitted to the Eastern Michigan University Department of Computer Science In Partial Fulfillment of the Requirements for the Master of Science in Computer Science. Ypsilanti, Michigan. (în format electronic la: <<http://www.emich.edu/compsci/projects/Master_Thesis_-_Benjamin_Speckmann.pdf>>, accesat ultima oară la 17 decembrie 2011)

² Pagina articolului: <http://www.gartner.com/it/page.jsp?id=1764714>

Un al doilea pas important ce trebui reamintit a fost în 2007 la fondarea consorțiului comercial Open Handset Alliance la care au aderat 48 de companii hardware, software și de telecomunicații, dispuse să ajute la dezvoltarea standardelor deschise pentru dispozitive mobile.³

În acest mod s-a dezvoltat platforma Android constituită dintr-un nucleu Linux, cu middleware, librării și API-uri scrise în C, precum și aplicații software ce rulează pe platforme ce includ librării compatibile cu Java, bazate pe Apache, o licență de tip free-software și open source, aspect ce îl face atractiv în rândul dezvoltatorilor.⁴

“Imaginația noastră este singura limită a ceea ce putem spera să realizăm în viitor.”⁵, constata inventatorul american Charles Franklin Kettering. În contextul societății actuale, înțelesul acestei mărturii se poate extinde în sensul că, ceea ce conduce la progresul tehnologiei și la dezvoltarea de software inovativ este finalmente, fantezia umană. Lucrarea de față are ca scop prezentarea unei aplicații care ilustrează caracterul social al omului - nevoia de a fi mereu în legătură cu familia și prietenii săi – avându-se în vedere facilitățile puse la dispoziție de sistemul de operare Android.

Aplicația MeetYou, așa cum am hotărât să o numesc, pune în legătură utilizatorii ei prin furnizarea locației unei persoane prietenilor, colegilor sau celorlalți membri ai familiei care folosesc această program. Un utilizator are posibilitatea de a fi notificat în cazul în care se află în apropierea unui prieten și de a lăsa pe alții să-i afle locația chiar și atunci când aplicația nu este deschisă prin trimiterea unui mesaj conținând un cuvânt cheie stabilit de către acesta.

Locația este precizată prin latitudinea și longitudinea locului în care se află telefonul mobil pe care rulează aplicația și poate fi vizualizată pe hărțile puse la dispoziție de serviciul Google Maps. Pe lângă aceasta, un utilizator se poate înregistra într-o anumită locație, pentru ca ceilalți prieteni să poată vizualiza și mai târziu locurile pe care le-a frecventat și eventual activitățile pe care le-a desfășurat în acele locuri. Una dintre caracteristicile interesante puse la dispoziție de acest program, este posibilitatea vizualizării locației prietenilor utilizând realitatea augmentată, lucru realizat cu ajutorul bibliotecii puse la dispoziție de aplicația mix Augmented Reality Engine (mixare).

³ Android (Sistem de operare), Wikipedia (în format electronic la: <<[http://ro.wikipedia.org/wiki/Android_\(sistem_de_operare\)](http://ro.wikipedia.org/wiki/Android_(sistem_de_operare))>>, accesat ultima oară la 17 decembrie 2011)

⁴ Android Developers: <<<http://developer.android.com/guide/basics/what-is-android.html>>>, accesat ultima oară la 18 decembrie 2011

⁵ <http://www.citatepedia.ro/citat.php?id=40965>

Android

Următoarele două subcapitole oferă o privire de ansamblu asupra platformei Android, atât din punct de vedere hardware, cât și software și au fost realizate utilizându-se ca referințe: lucrarea *Android Forensics - Investigation, Analysis, and Mobile Security for Google Android* scrisă de Andrew Hoog și publicată la editura Elsevier în 2011 [6], precum și pagina web destinată programatorilor Android [7]. Ultimul subcapitol descrie componența unei aplicații Android și a fost elaborată consultând materialul *Professional Android™ 2 Application Development* scris de Reto Meier și publicat la editura Wiley Publishing în 2010 [34].

Platforma Hardware

Android a fost proiectat pentru a fi compatibil cu o gamă largă de componente hardware. Aceasta constituie o caracteristică importantă, întrucât permite producătorilor să conceapă și să integreze elementele necesare dispozitivului Android, făcându-l atractiv atât în rândul programatorilor cât și al consumatorilor.

Fiind o platformă destinată dezvoltării de aplicații pentru telefoane mobile, în realizarea ei s-a avut în vedere conservarea energiei acumulatorului precum și gestionarea în mod eficient a resurselor de memorie existente.

Așa cum apare în [6], majoritatea dispozitivelor Android au utilizat încă de la început procesoare ARM, cunoscute a fi destul de puternice, raportat la consumul mic de energie. Cu toate acestea, unele corporațiile care au portat Android pe alte platforme au ales să folosească propriile lor CPU-uri (ex: Intel a utilizat propriul lor procesor Atom).

Pentru a nu suprasolicita procesorul, designerii de celulare au realizat o componentă separată care să gestioneze complexitatea comunicării mobile. Modem-ul baseband și radioul sunt sisteme hardware și software care furnizează dispozitivelor Android o conexiune la telefonie mobilă, lucru ce permite atât transfer de voce, cât și de date.

După cum se poate vedea în Figura 1.1, la telefoanele inteligente întâlnim două tipuri de memorie: memorie cu acces aleator (RAM) și memorie nevolatilă flash NAND. Conform [6], memoria RAM este folosită de sistem pentru încărcarea, executarea și manipularea diferitelor părți ale sistemului

de operare, iar cea flash pentru stocarea sistemului de operare precum și a datelor dorite de utilizator. Datorită limitării impuse de spațiul restrâns, cele două tipuri de memorie sunt combinate într-o singură componentă, cunoscută sub numele de pachet multicip (MCP).

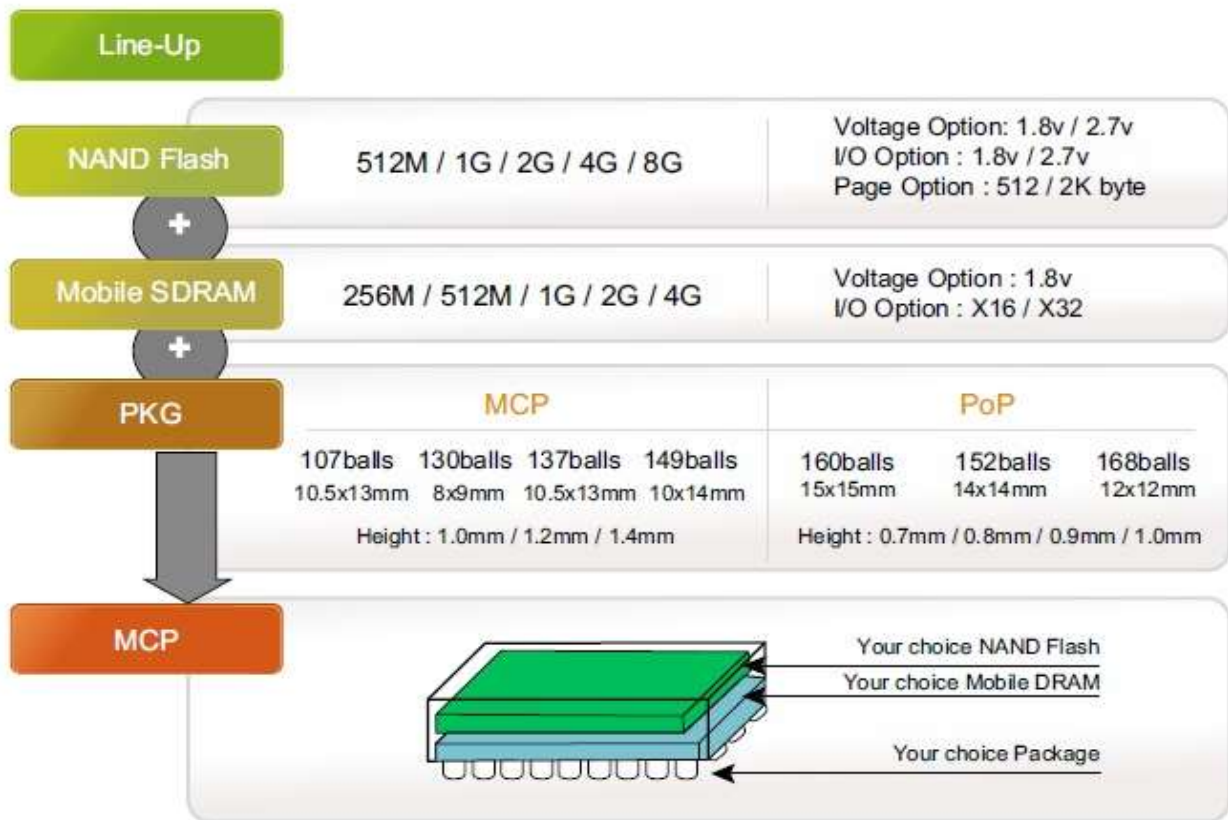


Figura 1.1 - Arhitectura MCP [6]

Una dintre cele mai interesante componente hardware este sistemul de poziționare globală (GPS) care este oferit gratis în pachetul de bază. Acesta nu identifică doar locația telefonului, folosindu-se de sateliții GPS, dar oferă și aplicațiilor posibilitatea de a accesa și de a se folosi de aceste informații.

Pe lângă rețeaua de telefonie mobilă existentă, Android permite conectarea la internet prin activarea opțiunii Wireless, cât și conectarea la alte telefoane sau dispozitive externe, prin activarea opțiunii Bluetooth.

Majoritatea telefoanelor prezintă un card de memorie detașabil care este cunoscut prin numele de card SD (Secure Digital). La fel ca memoria flash, cardurile SD sunt tipuri de memorie nevolatilă ce utilizează tehnologia flash NAND. Fiind concepute ca elemente portabile, acestea aderă la anumite specificații fizice, făcând posibil interoperabilitatea cu celelalte dispozitive. Acest aspect constituie una dintre notele distinctive vis-a-vis de telefoanele Apple, care nu prezintă card SD, ci doar memorie flash cu o capacitate de la 4 GB la 32 GB, după cum este relatat în [6]. Această alegere făcută de producătorii Android nu este doar o soluție mai ieftină ci și practică, acest card putând fi utilizat și de către alte telefoane.

Ecranul este interfața sistemului de operare pentru interacțiunea cu utilizatorul, fiind important nu doar pentru impactul pe care îl produce din punct de vedere vizual, cât și pentru răspunsul pe care îl oferă la atingere. Acesta este într-o continuă dezvoltare urmărindu-se creșterea rezoluției, a luminozității, precum și capacității de răspuns la stimulii externi.

Camera telefonului și-a depășit atribuțiile cunoscute de a face poze sau de a filma, utilizând celelalte componente hardware pentru a-și lărgi aria de utilitate. Majoritatea dispozitivelor combină funcționalitatea acestora pentru a înregistra pe lângă data și timpul la care s-a realizat o poză sau un video și coordonatele GPS la momentul utilizării acestora. Datele multimedia pot fi cu ușurință încărcate pe web și partajate cu alte persoane. Interesantă este utilizarea acestei componente în domeniul realității augmentate, camera putând fi folosită pentru a identifica un produs pe baza codului de bare sau a unei clădiri pe baza coordonatelor geografice sau a unor etichete.

Tastatura a fost și ea înlocuită cu una ce încorporează tehnologia touch screen. Aceasta se poate adapta după poziția telefonului și poate suporta mai multe limbi, existând și posibilitatea utilizării unor sisteme ce au rolul de a prezice cuvintele ce urmează a fi scrise.

Durata de viață a bateriei constituie încă o problemă datorită aplicațiilor existente ce sunt mari consumatoare de energie. Se dorește depășirea acestui neajuns, existând cercetări în acest sens, după cum apare în [6], ce urmăresc realizarea unor baterii din ce în ce mai puternice sau încărcarea bateriei fără utilizarea vreunui cablu, un exemplu fiind încărcarea în timpul mișcării.

Majoritatea dispozitivelor Android oferă suport pentru USB (Universal Serial Bus) cu ajutorul căruia telefoanele pot fi conectate la un calculator. Prin cablu se poate realiza atât încărcarea bateriei, cât și accesarea memoriei SD sau a consolei Linux de pe dispozitiv.

Android poate detecta și schimba poziția ecranului, în funcție de felul în care telefonul este ținut sau rotit, acest lucru fiind posibil datorită existenței unui accelerometru care detectează aceste schimbări. Noile versiuni suportă și un giroscop, care este mult mai sensibil decât un accelerometru, aspect ce a dus la dezvoltarea jocurilor pe această platformă.

Ultimele componente hardware sunt constituite de difuzorul și microfonul telefonului, care la fel ca și celelalte părți sunt perfecționate de la o versiune la alta. Unele dispozitive prezintă două sau trei microfoane, care combinate cu software-ul Android au abilitatea de a detecta și elimina zgomotul de fundal pentru a oferi o calitate mai bună a sunetului.

Sistemul de operare

Sistemul de operare este bazat pe nucleul Linux 2.6 care oferă software-ul necesar pentru a coordona atât partea hardware, cât și aplicațiile Android. Așa cum este ilustrat în Figura 1.2, funcțiile low-level includ administrarea energiei, a conexiunii wireless, a ecranului, a driverelor audio și a memoriei flash.

Peste kernel este construit un set de librării ce oferă funcționalitățile principale de care au nevoie atât dezvoltatorii, cât și utilizatorii. Acestea includ librării precum cele ce gestionează lucrul cu bazele de date (SQLite), cel ce furnizează utilizarea de socket-uri securizate bazate pe protocolul SSL, dar și librăria standard C.

Conform [7], librăriile principale sunt unite cu o mașină virtuală customizată Java, ce furnizează mediul de rulare pentru aplicațiile Android. Dezvoltatorii scriu și compilează programele cu kit-ul de dezvoltare Java oferit de companie Sun (Sun's Java Development Kit) și byte code-ul rezultat este transformat în fișiere de tip .dex ce sunt rulate apoi pe mașina virtuală Dalvik. Fiecare program rulează în procesul său propria lui instanță a mașinii virtuale Dalvik, care este concepută astfel încât mai multe multe instanțe ale acesteia să poată rula pe același dispozitiv într-un mod eficient.

Utilizarea acestor resurse este realizată prin intermediul unor API-uri și a unui application framework, cu care programatorii interacționează și care oferă acces acestora la metodele necesare în crearea aplicațiilor dorite.



Figura 1.2 – Structura sistemului de operare Android [7]

Componentele unei aplicații Android

În această secțiune sunt prezentate componentele slab cuplate ce formează o aplicație Android, modul cum acestea interacționează și felul cum acestea sunt legate prin intermediul fișierului manifest al programului din care fac parte.

Clasele de bază

Următoarele șase elemente furnizează scheletul pe baza căruia este construită o aplicație:

- **Activities** (Activități) Fiecare ecran din program este o extensie a clasei Activity. Activitățile utilizează elemente de vizualizare (*Views*) pentru a forma o interfață grafică care afișează informații și răspunde la acțiunile utilizatorului.

- **Services** (Servicii) Componentele serviciilor rulează în fundal și trimit date activităților ce permit actualizarea informațiilor expuse și declanșarea de notificări atunci când un eveniment așteptat are loc. Acestea pot rula chiar și atunci când activitățile nu sunt active sau vizibile.
- **Content Providers** (Furnizori de conținut) Furnizorii de conținut sunt utilizați pentru a administra și partaja bazele de date ale aplicației. Asta înseamnă că fiecare aplicație poate expune date ce pot fi utilizate ulterior de alte aplicații.
- **Intents** Cu ajutorul acestor resurse este posibilă difuzarea de mesaje la nivel de sistem, către o activitate sau un serviciu țintă, manifestându-se astfel intenția de a avea o acțiune efectuată.
- **Broadcast Receivers** (Consumatori de intenții difuzate) Dacă se crează sau se înregistrează un Broadcast Receiver, aplicația va începe să aștepte intențiile difuzate ce prezintă criteriul specificat și va răspunde la acțiunile acestora.
- **Widgets** Componente vizuale ce pot fi adăugate pe ecranul telefonului mobil
- **Notifications** (*Notificări*) Notificările sunt utilizate în scopul atenționării utilizatorilor asupra unui eveniment așteptat, fără ca activitățile curente să-și piardă focalizarea sau să fie întrerupte.

Fișierul Manifest

Fiecare proiect Android include un fișier manifest, `AndroidManifest.xml`, ce permite definirea structurii și meta datei aplicației, componentele și cerințele acesteia. Acesta include noduri pentru fiecare dintre componentele prezentate mai sus (activități, servicii ș.a.), filtre pentru intenții (*intent filters*) și permisiuni ce determină modul cum acestea interacționează. Fișierul manifest permite totodată specificarea de meta dată cu privire la aplicația curentă (iconițe și teme folosite), precum și noduri de nivel înalt ce includ setări de securitate, teste și cerințe hardware ale aplicației.

Manifestul este format dintr-o rădăcină reprezentată de eticheta `<manifest>` ce specifică prin atributul `xmlns:android` elementele de securitate ce sunt utilizate în cadrul fișierului, prin atributele `android:versionCode` și `android:versionName` versiunea aplicației, iar prin `android:package` pachetul principal al aplicației.

Următoarea listă oferă o perspectivă asupra nodurilor ce pot fi specificate în fișierul manifest și rolul acestora în construcția aplicației.

- ***uses-sdk*** marchează versiunea SDK ce trebuie să fie disponibilă pe un dispozitiv Android pentru ca aplicația să funcționeze precum și versiunea pe care s-a realizat și testat aplicația

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.facebook.meetyou"
  android:versionCode="1"
  android:versionName="1.0" >
```

- ***uses-configuration*** specifică o combinație dintre mecanismele de intrare suportate de aplicație, acestea făcând referire la tastatura disponibilă, la modul de navigare într-o fereastră, și la modul de atingere.

```
<uses-configuration android:reqTouchScreen=["finger"]
  android:reqNavigation=["trackball"]
  android:reqHardKeyboard=["true"]
  android:reqKeyboardType=["qwerty"]/>
```

- ***uses-feature*** ilustrează cerințele hardware ale aplicației și împiedică instalarea acesteia pe un dispozitiv ce nu respectă aceste condiții

```
<uses-feature
  android:name="android.hardware.camera.autofocus"
  android:required="false" />
```

- ***supports-screens*** enumeră dimensiunile ecranelor pe care aplicația le suportă

```
<supports-screens android:smallScreens=["false"]
  android:normalScreens=["true"]
  android:largeScreens=["true"]
  android:anyDensity=["false"] />
```

➤ **application** este un nod singular ce specifică meta data aplicației precum și componentele acesteia, constituind un container pentru noduri precum *activity*, *service*, *receiver* sau *provider*

- **activity** este un nod necesar pentru fiecare activitate ce rulează în interiorul aplicației, fiecare nod de acest tip având posibilitatea specificării prin intermediul nodului fiu `<intent-filter>` a elementelor intent ce pot lansa această activitate.

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

- **service** desemnează un serviciu din aplicație

```
<service
  android:label="@string/send_coordinates"
  android:name="service.SendCoordinates" />
```

- **provider** specifică fiecare din furnizorii de date ale aplicației

```
<provider android:permission="com.paad.MY_PERMISSION"
  android:name=".MyContentProvider"
  android:enabled="true"
  android:authorities="com.paad.myapp.MyContentProvider">
</provider>
```

- **receiver** are rolul de a înregistra un Broadcast Receiver, fără a fi nevoie de a lansa aplicația mai întâi

```
<receiver android:name="sms.SmsReceiver">
  <intent-filter>
    <action android:name=
      "android.provider.Telephony.SMS_RECEIVED" />
  </intent-filter>
</receiver>
```

➤ **uses-permission** stabilește permisiunile hardware de care are nevoie aplicația pentru a putea fi utilizată așa cum este așteptat și vor fi afișate utilizatorului înainte de a fi instalată

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

- **permission** este utilizată pentru restricționarea accesului unei alte aplicații la o componentă a sa. Pentru a avea acces la aceste elemente protejate, celelalte aplicații vor trebui să includă în manifestul lor noduri de tip `uses-permission`

```
<permission android:name="com.paad.DETONATE_DEVICE"
    android:protectionLevel="dangerous"
    android:label="Self Destruct"
    android:description="@string/detonate_description">
</permission>
```

- **instrumentation** oferă un cadru de testare pentru componentele aplicației la momentul execuției. Acestea oferă mijloace de monitorizare a aplicației și a interacțiunii cu resursele sistemului.

```
<instrumentation android:label="My Test"
    android:name=".MyTestClass"
    android:targetPackage="com.paad.aPackage">
</instrumentation>
```

Utilizarea de resurse externe

Mediul de programare Android oferă posibilitatea separării codului de celelalte resurse statice reprezentate prin imagini, șiruri de caractere, culori, animații sau teme, lucru ce permite o menținere, actualizare și administrare a aplicației mult mai eficientă.

Crearea de resurse

Resursele aplicației sunt stocate în directorul `res/` din ierarhia proiectului. Acest dosar este format din mai multe subdirectoare ce desemnează tipurile de resurse existente: `values` (valori), `drawable`, `drawable-hdpi`, `drawable-ldpi`, `drawable-mdpi`, (resurse imagini pentru afișări DPI înalte, medii sau joase), `layout` (aspect) și `menu` (meniu). Când are loc *build-ul* aplicației, aceste resurse vor fi compilate și incluse în pachetul aplicației, generându-se în același timp și un fișier `R.class` ce conține referințe la fiecare resursă inclusă în proiect.

Crearea de resurse valori

Valorile suportate includ șiruri de caractere, culori, dimensiuni, vectori de numere întregi sau vectori de șiruri de caractere și sunt păstrate independent în directoare separate.

Resurse String

Resursele de tip șir de caractere au ca etichetă elementul `<string>` și suportă stilizarea utilizând elemente specifice limbajului HTML⁶. Totodată este posibilă utilizarea de elemente *string* parametrizate și manipularea lor în cadrul codului utilizând metoda `String.format`.

```
<string name="remove_group" formatted="false">Are you sure you want to remove %1$s as your group?</string>
```

```
String message = String.format(getString(R.string.remove_group), group.getName());
```

Resurse Color

Pentru definirea unei resurse de tip culoare, este necesară utilizarea tag-ului `<color>`, specificarea numelui, precum și a codului potrivit utilizând una dintre notațiile:

- `#RGB`
- `#RRGGBB`
- `#ARGB`
- `#AARRGGBB`

unde A desemnează gradul de transparență, iar R, G, B valorile culorilor roșu, verde și albastru.

```
<color name="opaque_blue">#00F</color>
```

Dimensiuni

Dimensiunile sunt utilizate atunci când se dorește crearea unor constante în ceea ce privește frontierele unui *layout* sau dimensiunea unui font de scriere. Pentru specificarea unei resurse de dimensiune se utilizează eticheta `<dimen>`, urmat de valoarea și scala de măsurare a acesteia:

- `px` (pixelii unui ecran)
- `in` (inci)
- `pt` (puncte fizice)
- `mm` (milimetri)
- `dp` (pixeli independenți de densitate)
- `sp` (pixeli independenți de scală)

⁶ HyperText Markup Language

```
<dimen name="standard_border">5dp</dimen>
```

Stiluri și teme

Resursele pentru stiluri oferă posibilitatea realizării unei aplicații cu un aspect consistent realizat prin intermediul definirii de culori și fonturi după cum urmează:

```
<style name="BaseText">
    <item name="android:textSize">14sp</item>
    <item name="android:textColor">#111</item>
</style>
```

Elemente desenate (Drawables)

Resursele Drawable includ bitmaps, fișiere NinePatch, precum și componente complexe de tip Drawable ce ilustrează modul cum trebuie să se comporte un element vizual atunci când anumite acțiuni precum focalizarea sau atingerea au loc.

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true"
        android:drawable="@drawable/accept_pressed" /> <!-- apăsat -->
    <item android:state_focused="true"
        android:drawable="@drawable/accept_pressed" /> <!-- focalizat -->
    <item android:drawable="@drawable/accept" /> <!-- stare normală -->
</selector>
```

Layout

Resursele Layout permit decuplarea reprezentării vizuale a aplicației de modul cum utilizatorul acționează cu aceasta. Cel mai des mod de utilizare este definirea interfeței grafice a unei activități. Odată definit, aspectul unei activități poate fi setat în metoda onCreate prin apelarea funcției setContentView.

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

```

    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Hello World!"/>
    </LinearLayout>

```

Animații

Android suportă două tipuri de animație. Animațiile Tweened pot fi folosite pentru a roti, muta, întinde sau a face un element de vizualizare să dispară, iar animațiile Frame-By-Frame permit crearea unei animații ce afișează o serie de elemente de tip Drawable.

```

<animation-list
xmlns:android="http://schemas.android.com/apk/res/android"
android:oneshot="false">
    <item android:drawable="@drawable/rocket1" android:duration="500" />
    <item android:drawable="@drawable/rocket2" android:duration="500" />
    <item android:drawable="@drawable/rocket3" android:duration="500" />
</animation-list>

```

Meniuri

Meniurile reprezintă o altă formă de decuplare a interfeței grafice de modul de interacționare cu utilizatorul, fiind posibilă atât definirea de meniuri de activitate, cât și de context, fiecare meniu fiind memorat într-un fișier separat.

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menu_refresh"
        android:title="Refresh" />
    <item android:id="@+id/menu_settings"
        android:title="Settings" />

```

</menu>

Crearea de resurse pentru diferite limbi

Una dintre motivele imperioase de externalizare a resurselor este mecanismul dinamic de selectare a resurselor. Utilizându-se o ierarhie de directoare, se pot crea diferite valori corespunzătoare limbii și locației dispozitivului hardware utilizat. Android va face la momentul rulării o alegere dinamică a resurselor ce urmează a fi folosite, pe baza următorilor calificatori.

- **Codul țării și a rețelei mobile (MCC/MNC⁷)** O listă cu codurile disponibile poate fi accesată la adresa http://en.wikipedia.org/wiki/Mobile_Network_Code (e.g. mcc234-mnc20)
- **Limba și Regiunea** Limba este specificată prin 2 litere mici ISO 639-1 ce desemnează codul limbii, urmat opțional de regiune definită prin litera r, urmat de două majuscule ISO 3166-1-alpha-2 (en-rUS)

Rezumat

Acest capitol oferă o privire de ansamblu asupra platformei Android atât din punct de vedere hardware, cât și software. Deși unele componente variază de la un producător la altul, structura este comună pentru fiecare dintre aceste dispozitive.

Telefonul poate fi privit asemenea unui mini computer, prezentând un procesor de tip ARM și două tipuri de memorie, combinate într-o singură componentă, numită pachet multicip (MCP). Conexiunea la rețeaua mobilă este realizată de către modem-ul baseband și radioul existent, conectarea la internet prin activarea opțiunii Wireless, iar la alte telefoane sau dispozitive externe prin activarea opțiunii Bluetooth sau USB. Pe lângă acestea întâlnim o cameră performantă, un ecran touch screen și un sistem de localizare GPS.

Sistemul de operare este bazat pe nucleul Linux 2.6 și oferă software-ul necesar pentru a coordona atât partea hardware, cât și aplicațiile Android. Librăriile sunt accesate de programatori prin intermediul API-ului existent, iar rularea programelor se realizează pe mașinile virtuale Dalvik.

⁷ Mobile Country Code/Mobile Network Code

Din punct de vedere structural, o aplicație Android este compusă din două elemente principale: clasele ce descriu funcționalitatea programului și resursele constituite din aspecte, imagini, șiruri de caractere sau culori ce descriu interfața programului. Definirea structurii și metadatei aplicației, componentele și cerințele acesteia sunt precizate în fișierul manifest al aplicației, fișier fără de care o aplicație nu poate rula.

Capitolul 2

State of the Art

Aplicații de supravegheat persoane

În era digitală acaparată de Facebook, MySpace, Twitter și mai nou Google+, publicul larg, dar în special tinerii doresc să împărtășească cât mai multe informații din viața de zi cu zi cu prietenii sau familia lor. Observând această tendință, companiile specializate în crearea de aplicații pentru telefoane mobile s-au gândit să conceapă programe prin care datele partajate de utilizatori pot fi folosite într-un mod eficient.

Acest capitol oferă o imagine de ansamblu asupra aplicațiilor existente în această ramură și la a căror dezvoltare s-au folosit uneltele puse la dispoziție de Android.

Google Maps

Google Maps este cunoscut pentru funcționalitățile pe care le are: vizionarea de hărți 2D, 3D sau street view, consultarea traficului în timp real în anumite zone, oferirea de indicii în privința alegerii unui drum pentru a ajunge la o anumită destinație sau chiar utilizarea unui sistem de navigație GPS cu îndrumare vocală, dar ceea ce este interesant în cazul de față este opțiunea Latitude prin care utilizatorul îți poate înregistra locația și o poate partaja cu prietenii săi.

Vizualizarea prietenilor se realizează pe hărțile 2D, așa cum poate fi observat și în Figura 2.1. Utilizatorii au opțiunea de a controla nivelul la care prietenii au acces la informațiile ce țin de locație, precum și posibilitatea de a ascunde aceste informații atunci când este necesar. În cazul în care un prieten nu posedă un smartphone, dar are cont Gmail și acces la internet, acesta poate vizualiza datele persoanelor ce îl interesează.

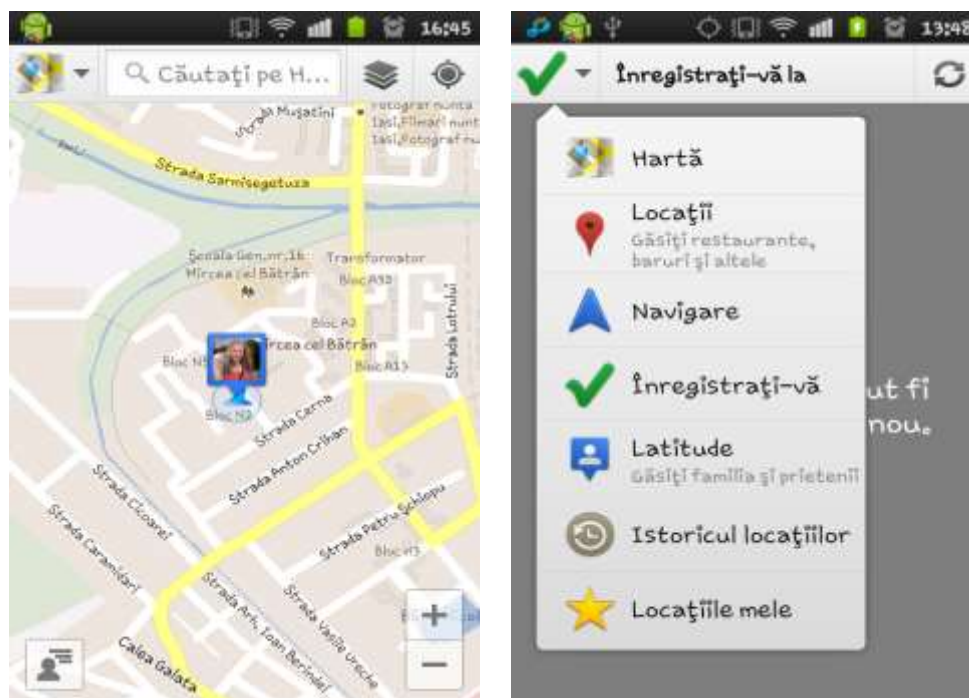


Figura 2.1 - Google Maps Latitude

Gypsii

Această aplicație permite crearea de noi puncte pe harta disponibilă și etichetarea acestora cu poze, filme și mesaje. Totodată este posibil conectarea la contul de Facebook sau de Twitter și partajarea informațiilor referitoare la locația curentă. Așa cum poate fi remarcat în Figura 2.2, utilizatorii pot împărtăși impresii sau opinii referitoare la locurile pe care le frecventează, prietenii având posibilitatea de a explora comentariile și recomandările acestora.

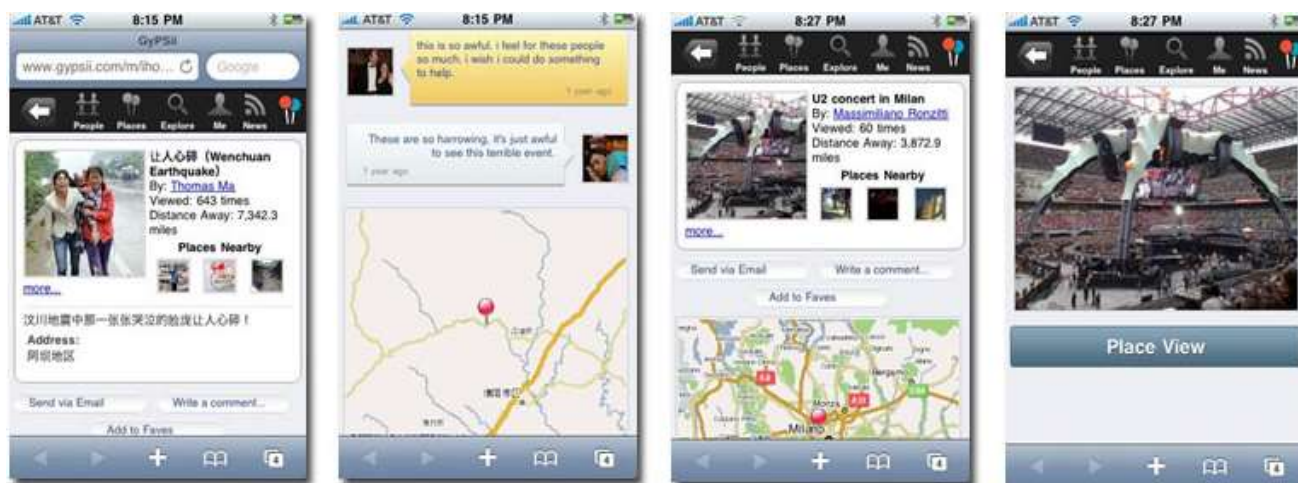


Figura 2.2 - Gypsii [10]

Pagina de noutăți prezintă noile informații postate de ceilalți prieteni, cea de persoane oferă date privitoare la localizarea tuturor utilizatorilor, iar cea pentru prieteni filtrează userii, rămânând doar informațiile corespunzătoare celor din listă de contacte. Faptul că o persoană are acces la locația altor prieteni, face posibil ca la un moment dat două contacte să stabilească o întâlnire, atunci când observă că sunt în același perimetru.

LOCiMobile

LOCiMobile oferă două aplicații pentru platforma Android din care clienții pot alege în funcție de ceea ce își doresc. Conform paginii oficiale a acestei companii⁸, GPS Tracking prezintă posibilitatea localizării altor persoane, acces la rețelele de socializare Facebook și Twitter, scrierea de mesaje către unul sau mai mulți prieteni și încărcarea de poze. Spre deosebire de aceasta, Tracking oferă beneficiile de a localiza și a oferi informații în timp real despre un contact sau un grup de contacte, utilizând atât aplicația instalată pe mobil cât și pagina web a acesteia.

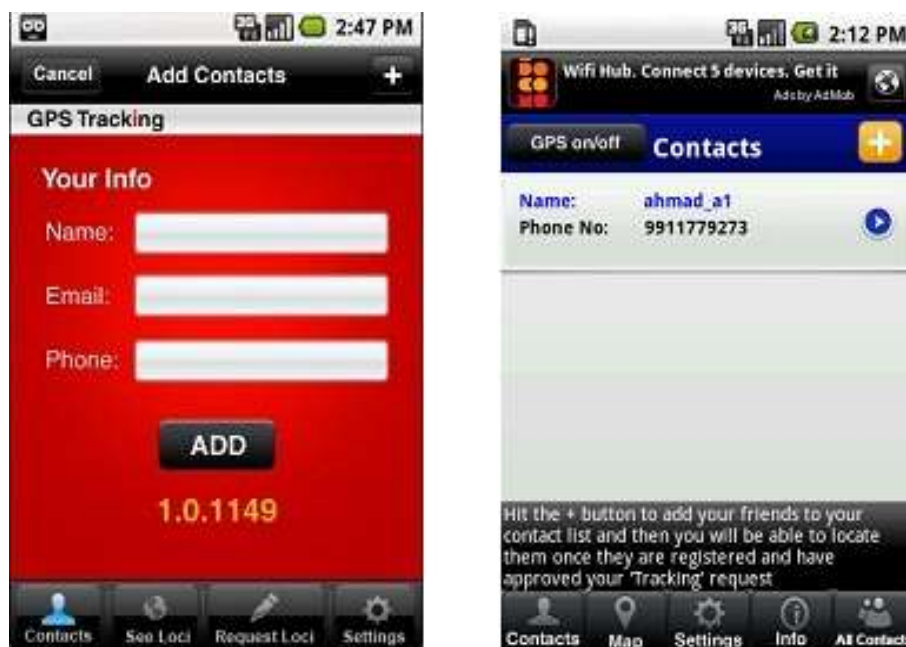


Figura 2.3 Aplicațiile LOCiMobile [22][23]

⁸ Pagina oficială LociMobile: <<<http://www.locimobile.com/apps/>>>, accesată ultima oară la 19 decembrie 2011

Life360

Life360 a fost conceput ca un mod de a ține familia în legătură, transformând telefoanele mobile în dispozitive de localizare. Folosind tehnologia GPS, cea Wireless precum și rețeaua telefonică, utilizatorii își pot împărtăși locația sau pot verifica locația celorlalți membrii ai familiei.

Utilizatorul are posibilitatea de a-și anunța familia atunci când este într-un loc sigur și de a realiza un apel de panică, atunci când se află în pericol. Există un canal de comunicare cu ceilalți membri ai familiei unde un utilizator își poate înregistra noua locație sau unde poate posta un mesaj pentru toți cei care au acces la acest canal, aspect redat în Figura 2.4.

Programul ține și o evidență a perioadei care a trecut de când s-a făcut ultima înregistrare a locației, precum și a tuturor punctelor de pe hartă prin care a trecut acea persoană.

Fiind concepută ca o aplicație care să ofere siguranță persoanelor care o folosesc, aceasta furnizează și informații în legătură cu spitalele și secțiile de poliție aflate în apropierea zonei în care un utilizator s-a înregistrat.

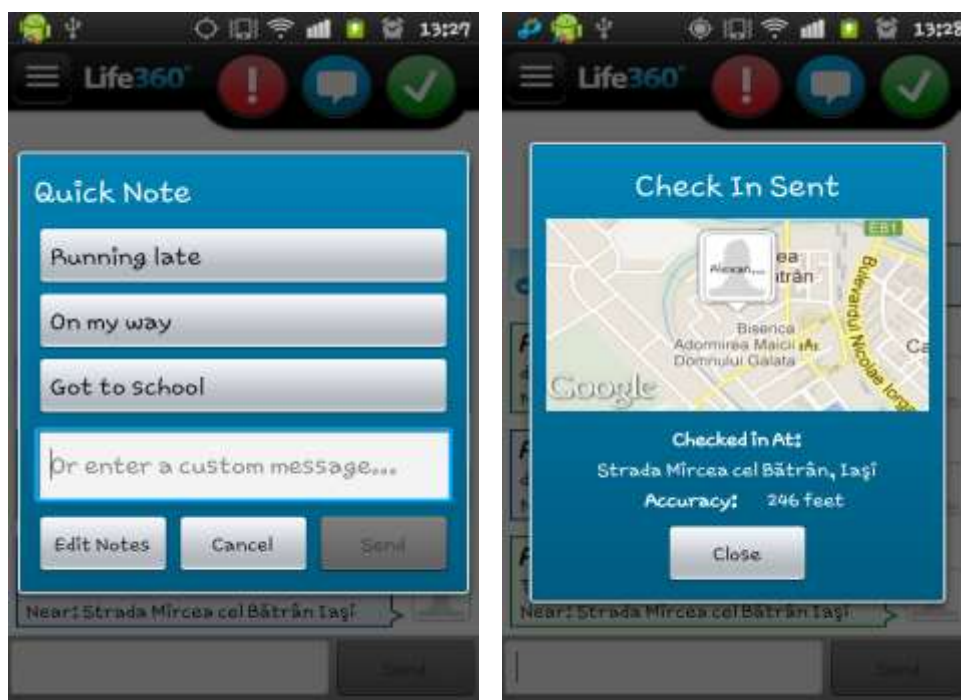


Figura 2.4 - Life360

Match2Blue

Spre deosebire de celelalte aplicații ce au rolul de a pune în legătura utilizatorii cu familia sau prietenii lor, Match2Blue își ajută clienții să cunoască oameni noi cu care pot interrelaționa. Cum se poate vedea în Figura 2.5, pagina principală enumeră persoanele conectate, ordonate în funcție de distanța la care se află față de utilizatorul curent. Prin încărcarea de poze și utilizarea unor mesaje interesante, aceștia încep să socializeze, formându-se astfel noi legături de prietenie.



Figura 2.5 – Match2blue [24]

Glympse

Acest program este eficient atunci când o persoană dorește să partajeze cu alți oameni drumul pe care îl parcurge într-o perioadă prestabilită de timp. Plusul pe care îl oferă este constituit de faptul că nici unul dintre utilizatori nu este nevoit să își facă cont și doar cel care vrea să împărtășească aceste informații, trebuie să aibă instalată aplicația pe telefonul mobil, cealaltă persoană putând viziona drumul parcurs pe orice browser web (Figura 2.6), accesând link-ul primit pe adresa de e-mail.

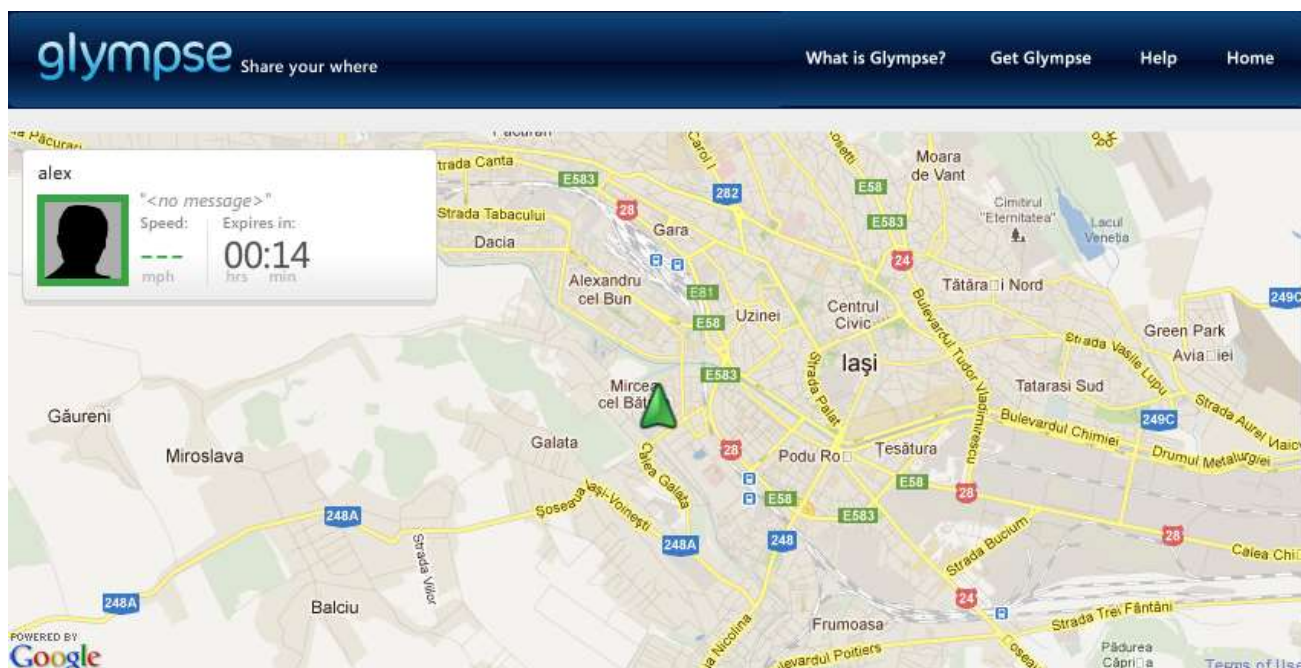


Figura 2.6 - Glympse

Platforme de dezvoltare pentru Realitatea Augmentată

Nu cu mult timp în urmă, realitatea augmentată reprezenta un experiment tehnologic foarte rar folosit în afara laboratoarelor de specialitate, întrucât erau necesare cunoștințe și tehnică de nivel înalt. Acum, datorită avansărilor făcute în hardware-ul telefoanelor mobile, tehnologia AR este mult mai accesibilă atât pentru utilizatori cât și pentru dezvoltatori.

Această secțiune se va focaliza pe analizarea framework-urilor care permit un anumit nivel de customizare, permițând publicarea propriilor informații, și mai puțin asupra aplicațiilor de sine stătătoare.

Look!

Look! Este o platformă pentru realitatea augmentată, specifică sistemului de operare Android și creată sub licența GPLv3, ce integrează facilități precum: desenarea de grafice 2D sau 3D, interacțiunea cu obiecte virtuale, localizarea de entități în exteriorul sau interiorul clădirilor, integrarea cu servicii remote și persistența datelor.

După cum este ilustrat pe pagina de documentație a acestui framework⁹, sistemul de localizare specific spațiilor închise cuprinde următoarele module:

- Sistemul de Localizare primar ce utilizează semnalele wireless și returnează o locație.
- Sistemul de Localizare Secundar bazat pe senzorii telefonului mobil ce detectează mișcarea dispozitivului în spațiu.
- Integrarea sistemului primar cu cel secundar.

Spre deosebire de celelalte platforme, aceasta nu oferă nicio aplicație de sine stătătoare.

Unifeye SDK Mobile

Această librărie este oferită spre vânzare de către firma germană Metaio și oferă dezvoltatorilor, după cum se poate vedea în Figura 2.7, facilități precum urmărirea, captarea sau redarea de obiecte. În acest scop, aceștia pot folosi atât o aplicație standard bazată pe configurarea de fișiere XML, cât și API-ului de nivel înalt (scris în limbajul de programare C), prin apelul de funcții ale puse la dispoziție de acesta.

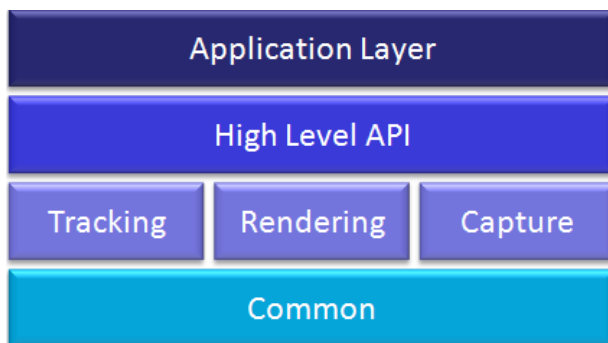


Figura 2.7 - Structură Bibliotecă Unifeye [14]

Cea mai importantă aplicație construită pe această platformă este constituită de Junaio care inițial a fost concepută ca un browser pentru rețelele sociale ce au la bază realitatea augmentată și în care unii utilizatorii pot încărca obiecte 3D în lumea virtuală, iar alții le pot adăuga la scene 3D.

⁹ Localizarea în interiorul clădirilor cu Look!: <<<http://www.lookar.net/2011/07/localizacion-en-interiores/>>>, accesată ultima oară la 28 decembrie 2011

În versiunea 2.0, Junaio a renunțat la crearea de scene 3D și a devenit propriul lui augmented reality browser, oferind oamenilor posibilitatea accesării barurilor, restaurantelor sau al altor locații, devenind și primul browser AR care suportă aducerea imaginilor la viața prin recunoașterea de imagini.

Junaio 3.0 duce caracteristicile deja existente la un alt nivel. Noua funcție „SCAN” scanează obiecte precum: picturi, coduri QR și chiar codurile de bară ale produselor. Focalizând cu ajutorul camerei lucrurile din jur, se pot obține informații din canalele Junaio, din platformele partenerilor săi sau din alte baze de date. Astfel, atunci când Junaio recunoaște un cod de bare, combină și afișează datele obținute conducând utilizatorul către un website, un magazin online sau către alte informații necesare, lucru vizibil în Figura 2.8. Acest lucru este posibil prin existența unui web browser intern care permite apăsarea iconiței unui punct de interes (POI) și accesarea unui website fără ca aplicația Junaio să fie părăsită.¹⁰



Figura 2.8 – Junaio [27]

¹⁰ "Scan the World" with the new junaio 3.0 Augmented Reality Browser, (în format electronic la: <<<http://www.metaio.com/press/press-release/2011/scan-the-world-with-the-new-junaio-30-release/>>>), accesat ultima oară la 3 ianuarie 2012).

Mixare

Mixare este un alt browser publicat gratuit sub licența GPLv3, care este disponibil atât ca o aplicație de sine stătătoare, cât și ca librărie pentru integrarea în implementări proprii, permițându-se astfel afișarea locurilor de interes, într-un cadru specific realității augmentate, aspect ilustrat în Figura 2.9.

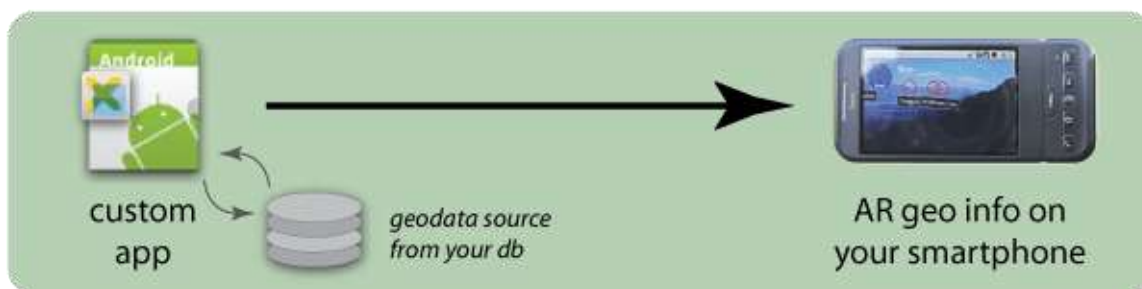


Figura 2.9 - Afișarea punctelor geografice utilizând Mixare [17]

Wikitude

Wikitude este un software pentru realitate augmentă, realizat pentru telefonie mobilă, ce a fost dezvoltat de către firma austriacă Wikitude GmbH (Mobilizy GmbH) și scos pe piață în octombrie 2008 ca un produs gratuit. Acesta constituie prima aplicație care folosește localizarea spațială pentru a obține o realitate augmentată.



Figura 2.10 Wikitude [25]

Wikitude explorează diverse puncte de interes aparținând canalelor Wikipedia, Youtube, Twitter, Flickr, Starbucks și altor pagini web cu un conținut captivant pentru utilizator. Wikitude oferă și o platformă deschisă pentru dezvoltatorii software care se pot bucura în totalitate de funcționalitățile oferite abia după înregistrare, când filigranul din versiunea demo dispare și este posibilă publicarea propriei implementări.

Layar

Layar, un browser pentru realitatea augmentată, ce aparține unei companii din Amstardam, este cea mai cunoscută și mai răspândită aplicație pentru realitatea augmentată realizată pentru dispozitive smartphone după cum arată statisticile de pe Android Market¹¹.

Browser-ul Layar Reality arată ceea ce este în apropierea utilizatorului prin afișarea informațiilor digitale (layers) în timp real peste realitatea lumii văzută prin cameră. Acesta funcționează prin utilizarea unei combinații dintre cameră, compas și datele GPS pentru identificarea locației utilizatorului și a suprafeței apropiate acestuia, preluarea datelor bazate pe acele coordonate geografice și suprapunerea acestor informații peste suprafața camerei.



Figura 2.11 – Layar [19]

Pe lângă adăugarea punctelor de interes (POI), Layar permite și adăugarea de obiecte 3D. Această caracteristică este foarte utilă pentru a arăta lumii ceea ce nu există încă sau ceea ce nu mai există. Spre exemplu se poate reda o clădire romană, într-un loc unde nu există decât niște ruine sau planul unei clădiri ce este în plan pentru a fi construită. Aceste obiecte 3D pot fi legate de pagini web, astfel încât prin accesarea lor, utilizatorul poate citi mai multe informații care îl pot interesa. Totodată oferă suport audio și video, permite setarea unor alerte care să se declanșeze în apropierea unei locații prestabilite precum și object tracking sau image recognition.

După cum este prezentat pe pagina oficială¹², portofoliul Layar prezintă pe lângă browser alte trei produse și anume: Layar Player, SDK ce face posibil integrarea de layers AR într-o aplicație proprie, dar care este disponibil doar pentru sistemul de operare iOS, Layar Connect, API pentru serverul Layar ce oferă posibilitatea construirii unui sistem de management pentru layers și Layar Vision ce utilizează tehnicile computer vision pentru a augmenta obiectele în lumea fizică.

LibreGeoSocial

LibreGeoSocial este o comunitate ce lucrează la crearea unui framework open source pentru o rețea socială bazată pe aplicații AR. Un browser prototip este disponibil pentru descărcat, iar dezvoltatorii pot accesa atât partea de client cât și partea de server, construit pe baza unui puternic API, ce oferă ca funcționalități urmărirea prin localizare precum și recunoașterea de imagini. Suportul pentru rețeaua socială și etichetarea este de asemenea inclusă în plugin-ul browser-ului.¹³

Sistem funcționează atât în interior, cât și în exterior, printr-o localizare oferită de BIDI. Nodurile rețelei sociale sunt localizate și poziționate la diferite altitudini. Acestea pot fi accesate clasic dintr-o listă sau dintr-o interfață AR, unde nu se pot doar vedea, ci este posibil și adăugarea altor etichete.

O altă funcționalitate este dată de existența unei alarme care se declanșează atunci când utilizatorul este în apropierea unor locuri ce au fost specificate la crearea lor.

¹² Pagina oficială Layar: <<<http://www.layar.com/>>>, accesată ultima oară la 29 decembrie 2011

¹³ Butchart, B., Augmented Reality for Smartphones, (în format electronic la: <<http://observatory.jisc.ac.uk/docs/AR_Smartphones.pdf>>, accesat ultima oară la 19 septembrie 2011)



Figura 2.12 – LibreGeoSocial [26]

Rezumat

Nevoia oamenilor de fi în legătură cu prietenii sau familia lor este vizibilă în entuziasmul cu care aceștia folosesc aplicațiile și website-urile de socializare. Acest lucru a determinat ca odată cu dezvoltarea dispozitivelor mobile, să se dezvolte și această ramură, programele de socializare fiind portate pe platforma telefoanelor inteligente și combinate cu alte tehnologii inovatoare precum realitatea augmentată.

În acest sens s-au creat aplicații prin care părinții își pot supraveghea într-un mod eficient copiii (Life360), prin care tinerii pot interrelaționa cu prietenii lor, pot cunoaște noi persoane și pot schimba impresii în legătură cu locurile pe care le frecventează (Gypsii, LOCiMobile, Match2blue) sau prin care mai multe persoane pot partaja drumul pe care îl parcurg într-o anumită perioadă de timp (Glympse).

Prin perfecționarea camerei foto și introducerea sistemului GPS, aplicațiile ce folosesc realitatea augmentată au devenit din ce în ce mai atrăgătoare în rândul programatorilor, care începând cu utilizarea punctelor geografice pentru a crea o lume augmentată (Mixare, Look!), s-au extins prin dezvoltarea tehnologiei de recunoaștere de obiectelor 3D (Layar) precum și produselor, prin scanarea codului de bare al acestora (Junaio).

Capitolul III

Descrierea aplicației MeetYou

Aplicația prezentată în această lucrare este alcătuită din trei module corespunzătoare celor trei nivele ale modelului arhitectural client – server *Three-tier* (3-tier architecture¹⁴), unde elementele de interfață, de procesare și datele sunt separate. Nivelul de prezentare este constituit de aplicația Android cu ajutorul căreia vor fi expuse informațiile referitoare la utilizatori, nivelul de date este reprezentat de baza de date în care sunt stocate aceste informații, iar nivelul logic reprezintă legătura dintre cele doua nivele și este realizată cu ajutorul serviciilor web și a serviciilor Android.

Nivelul logic – Tehnologiile utilizate

Android Location Service

Informațiile privind localizarea spațială devin din ce în ce mai importante în lumea dezvoltării aplicațiilor mobile. Capacitatea de a prelua și de a furniza cu ușurință aceste date este utilizată cu succes pe platformele mobile din zilele noastre. Android oferă această funcționalitate utilizând serviciile de localizare (*location service*). Acestea oferă acces la facilități ce pot fi folosite pentru a determina poziția curentă. În Android, aceste facilități sunt cunoscute sub numele de *location providers* și fiecare dintre acestea au propriul lor set de puncte forte și puncte slabe. Următoarea secțiune descrie cum fiecare din aceste metode funcționează, informațiile fiind extrase din lucrarea „Professional Android Sensor Programming” scrisă de Greg Milette și Adam Stroud, publicată la editura Wrox în iunie 2012 [31].

¹⁴ http://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture

GPS Provider

Termenul GPS face referire la întreg sistemul format din sateliții ce orbitează în jurul planetei, receptorii, precum și stațiile de control care monitorizează procesul de stabilire a poziției curente. Sistemul de poziționare globală este format din 27 de sateliți ce se rotesc în jurul Pamântului și transmit informațiile către receptori. Fiecare satelit urmează o cale prestabilită, asigurând astfel ca la orice moment cel puțin 4 sateliți să fie disponibili în orice punct de pe glob. Figura 3.1 înfățișează o constelație a sateliților GPS.



Figura 3.1 Sateliții GPS [30]

Conform [31], fiecare satelit din constelație transmite continuu două tipuri de date: efemeride și almanah. Datele almanah includ informații orbitale despre fiecare satelit, precum și despre starea întregului sistem, și sunt transmise receptorului GPS. Datele efemeride conțin date despre starea satelitului, data și ora curentă. Prin cele două tipuri de informații este stabilită locația curentă, acuritatea acestui calcul fiind furnizată de datele efemeride.

Pentru a furniza locația sa, un receptor trebuie să fie în măsură să determine distanța de la el către ceilalți sateliți. Acest lucru este posibil prin utilizarea datelor efemeride ce includ poziția precum și timpul la care a avut loc transmisia. Folosind aceste distanțe, receptorul GPS poate triangula locația sa curentă. Mai exact, punctul de la care toate sferele se intersectează este poziția receptorului. Un minim de trei sateliți este necesar pentru a determina o locație bidimensională (latitudine și longitudine), ceilalți ajutând la calculul altor informații precum altitudinea.

Deși GPS este util în determinarea locației curente, acesta prezintă anumite dezavantaje, în special pe platformele telefoanelor mobile, unul fiind constituit de timpul în care se poate calcula poziția curentă. Înainte ca locația să poate fi calculată, trebuie să fie determinați sateliții ce ar trebui să ofere informațiile necesare, proces ce poate fi destul de lent. Un alt aspect este constituit de faptul că receptorul GPS are nevoie de o cale directă către sateliții GPS. Acest lucru înseamnă că receptorii GPS nu pot lucra în spațiile închise, dar și că ar putea exista probleme în cazul în care cerul nu ar fi vizibil. Acest aspect este afectat și de faptul că un telefon mobil nu poate conține unde radio de putere mare.

Network Provider

În Android, localizarea bazată pe rețea poate fi realizată în două moduri: utilizând turnurile celulare, precum și prin utilizarea informațiilor oferite de rețeaua wireless.

Utilizarea punctelor de acces wireless

Potrivit [31], detectarea locației se realizează prin scanarea rețelelor wireless disponibile. Informația ce include adresele MAC¹⁵ ale punctelor de acces Wi-Fi, precum și puterea semnalului primit de la aceste puncte este colectată mai întâi de către telefoanele mobile. Dispozitivul realizează apoi o interogare la serviciul de localizare Google (care este diferit de serviciul de localizare Android) de la care obține poziția curentă.

Două dintre beneficiile pe care le oferă localizarea prin Wi-Fi se referă la faptul că este posibilă depistarea poziției curente în spațiile închise, precum și la faptul că de cele mai multe ori timpul și consumul de baterie este mai mic decât la GPS.

Așa cum reiese din [31], această facilitate prezintă și dezavantaje. În primul rând, pentru a determina locația curentă, trebuie să existe puncte de acces disponibile. În plus, rețelele trebuie să prezinte un identificator uni SSID, ipsa acestuia precum și terminarea acestuia cu „_nomap” nefăcând posibilă partajarea informațiilor cu serviciul de localizare Google.

¹⁵ Media Access Control

Adițional, schimbarea locației punctelor de acces Wi-Fi poate cauza inexactități în datele privind poziția curentă. Dacă un utilizator, care deține o rețea wireless în propria casă, ar pleca cu dispozitivul în vacanță, serviciul de localizare ar putea determina greșit poziția curentă, bazându-se pe informațiile existente în sistem, ce au fost recepționate în trecut. Cu toate că serviciul de localizare permite schimbarea adresei punctelor de acces, acest lucru nu poate fi realizat și manual de către utilizator.

Utilizarea turnurilor de telefonie mobilă

Pentru a funcționa corect, un dispozitiv celular trebuie să fie în contact cu un turn de control. În timp ce telefonul este în mișcare, este posibil conectarea la alte turnuri atunci când puterea semnalului de la un turn de control de care se apropie devine mai puternică. Cunoașterea identificadorului unic al turnurilor la care a fost sau este conectat, poate oferi o perspectivă asupra zonei unde dispozitivul se află, în cazul în care adresa acestor turnuri celulare este cunoscută.

La fel ca și în cazul rețelelor wireless, Android lucrează cu serviciul de localizare Google pentru a asocia fiecărui turn celular o locație. Pentru acest caz, informațiile includ atât date despre turnul la care este conectat momentan, precum și informații despre ultima locație furnizată de GPS.

Atunci când un dispozitiv are nevoie să găsească locația curentă, trimite informații privind turnul celular la care este conectat, precum și despre cele la care a fost conectat serviciul Google. Cu cât numărul de identificatori trimiși este mai mare, cu atât și acuritatea informațiilor va fi mai mare. Determinarea locației nu este posibilă în cazul trimiterii unui singur ID celular.

Determinarea poziției unui telefon Android

Pentru dezvoltatorii de aplicații pentru telefoanele mobile care doresc să folosească informații ce privesc localizarea, Android oferă un API robust ce se folosește de tehnologiile prezentate mai sus pentru a obține datele dorite. Majoritatea claselor utilizate în acest scop se găsesc în pachetul *android.location* și includ membri precum: *LocationManager*, *LocationProvider*, *Location*, *Criteria*, *LocationListener*.

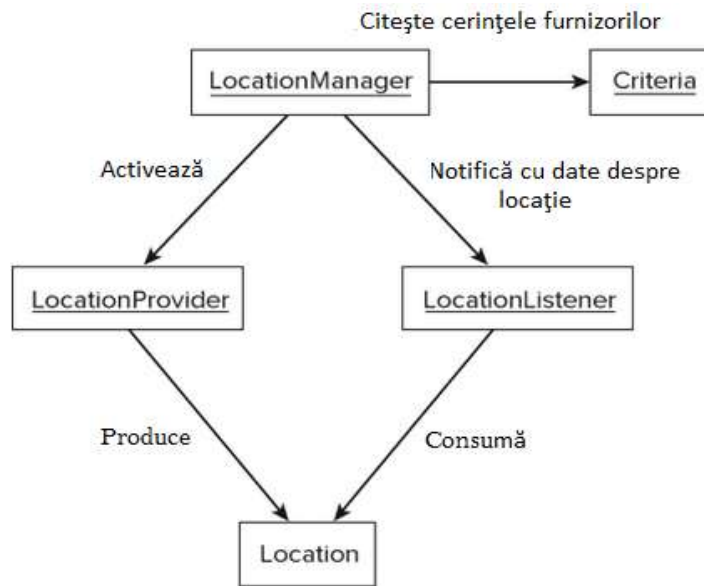


Figura 3.2 Componentele Android pentru localizare [31]

LocationManager

Location Manager este un serviciu la nivel de sistem. Serviciile la nivel de sistem sunt obținute direct de la context, utilizându-se numele acestora, nefiind posibilă instanțierea de obiecte. Acesta oferă informații privind starea curentă a sistemului de localizare, date ce includ furnizorii disponibili, furnizorii activi, precum și informații privind statusul GPS. Prin LocationManager se poate specifica timpul la care se dorește actualizarea locației curente și se poate obține efectiv ultima actualizarea a acesteia.

```

private Location getLocationByProvider(Context context, String provider) {
    Location location = null;
    LocationManager locationManager = (LocationManager) context
        .getSystemService(Context.LOCATION_SERVICE);
    if (!isProviderSupported(locationManager, provider))
        return null;
    location = locationManager.getLastKnownLocation(provider);
    return location;
}
  
```

Înainte de a putea utiliza Location Manager trebuie să adăugăm câteva permisiuni în fișierul manifest al proiectului pentru a sprijini accesul la componentele hardware necesare. În acest caz sunt necesare permisiunile pentru localizarea fină și grosieră.

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

Location Provider

Clasa *LocationProvider* este o abstractizare pentru sursele de la care pot proveni datele privind localizarea, surse ce au fost prezentate anterior. Deși fiecare dintre acestea generează informații în mod diferit, modul de comunicare cu aplicația este același.

Location

Location este clasa care încapsulează datele propriu-zise privind locația curentă a telefonului mobil și pune la dispoziție caracteristici precum latitudinea, longitudinea și altitudinea la care se află dispozitivul Android.

Criteria

O aplicație poate utiliza *Criteria* pentru a interoga *LocationManager* în scopul obținerii furnizorilor ce pot oferi anumite date de interes. Această clasă este utilă atunci când programatorul este concentrat mai mult asupra unor caracteristici pe care calculul obținerii locației trebuie să îi respecte și nu asupra furnizorilor folosiți.

LocationListener

Interfața *LocationListener* conține o serie de metode *callback* care sunt declanșate atunci când are loc schimbări în ceea ce privește locația curentă a dispozitivului (*onLocationChange*) sau starea serviciului de localizare (*onProviderEnabled*, *onProviderDisabled*, *onStatusChanged*). *LocationManager* permite ca o aplicație să înregistreze și să anuleze înregistrarea unei clase ce implementează *LocationListener* și care are ca rol procesarea informațiilor privind noua locație.

Google Maps

Pentru a putea afișa punctele găsite de *LocationManager* se pot utiliza clasele din biblioteca externă *Google Maps*. Pentru aceasta, proiectul construit trebuie să fie de tip Google APIs, o versiune standard Android neavând posibilitatea referențierii claselor necesare. Totodată fișierul manifest trebuie să specifice utilizarea acestei biblioteci și să ofere acces la internet pentru obținerea copiei necesare.

```
<uses-library android:name="com.google.android.maps" />
```

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

Componentele Google Maps

MapView

Clasa MapView este unealta care afișează harta în aplicația Android. Fișierul .xml ce specifică aspectul activității în care harta va fi încărcată include un element de tip *com.google.android.maps.MapView* cu atributul *android:apiKey*. Cheia este necesară pentru a avea acces la API-ul *Google Maps*, iar pentru a o obține este nevoie înregistrarea la serviciul *Google Maps*, urmând pașii prezentați la adresa <https://developers.google.com/maps/documentation/android/mapkey>.

OverlayItem

OverlayItem este un obiect ce marchează un element pe hartă. Acesta este un container format dintr-un marcator, un titlu, precum și un obiect de tip *GeoPoint* ce memorează locația – latitudine, longitudine, acuritate.

ItemizedOverlay

ItemizedOverlay ține lista de obiecte de tip *Overlay* și descrie modul cum acestea vor fi afișate. Întrucât *ItemizedOverlay* este o clasă abstractă, aplicația trebuie să conțină o clasă ce implementează metodele definite aici.

KML Format

Formatul KML¹⁶ este folosit pentru a expune date geografice într-un explorator pentru Pământ, așa cum sunt: *Google Earth*, *Google Maps* și *Google Maps* pentru mobil. KML prezintă o structură xml și trebuie realizat întocmai cu modelul listat în KML Reference¹⁷.

Pentru a se obține mulțimea de coordonate corespunzătoare drumului rutier dintre două puncte de pe glob, se poate realiza o cerere la adresa: <http://maps.google.com/maps?f=d&hl=en&saddr=slat,slng&daddr=dlat,dlng&ie=UTF8&0&om=0&output=kml>, unde slat, slng reprezintă latitudinea și longitudinea punctului sursă, iar dlat, dlng reprezintă latitudinea și longitudinea punctului destinație. Fișierul .kml rezultat este mai apoi parsat, utilizându-se o instanță a parserului *SAXParser* ce va lua ca parametru un handler, ce specifică modul cum informațiile vor fi extrase. Punctele obținute sunt transformate în elemente *GeoPoint* și pasate unui obiect ce moștenește proprietățile clasei abstracte *com.google.android.maps.Overlay*, pentru a fi afișate pe suprafața unei hărți.

JSONObject

Potrivit Wikipedia, JSON este un acronim în limba engleză pentru JavaScript Object Notation și este un format de reprezentare și interschimb de date între aplicații. Acesta este un format text, inteligibil pentru oameni, utilizat pentru reprezentarea obiectelor și a altor structuri de date și este folosit în special pentru a transmite date structurate prin rețea.

Obiectele JSON sunt construite în perechi de tip cheie:valoare. Elementele obiectului sunt separate prin virgule, fiecare obiect este delimitat cu ajutorul acoladelor, iar un șir de obiecte este memorat între paranteze pătrate.

Este un standard de scriere mai economic decât XML în ceea ce privește mărimea datelor, dar mai puțin lizibil decât acesta, limitat la valori text și valori numerice, nefiind posibilă memorarea informațiilor în format binar.

¹⁶ Keyhole Markup Language

¹⁷ KML Reference (în format electronic la adresa <<<https://developers.google.com/kml/documentation/kmlreference>>>, accesat ultima oară la 30 mai 2012)

În cazul de față, obiecte de tip JSON au fost folosite la schimbul de informații dintre aplicație și baza de date SQL, între aplicație și Facebook, pentru logarea și validarea conturilor utilizatorilor, și pentru afișarea prietenilor utilizând tehnologia realității augmentate.

JSONObject și baza de date

Așa cum este afirmat în unul din tutorialele IBM¹⁸, există o tendință răspândită în a folosi obiecte javascript pentru a transfera o serie de linii dintr-o bază de date către un vector de obiecte dintr-o aplicație, unde fiecare obiect reprezintă o linie și fiecare proprietate o coloană din baza de date (și viceversa).

Atunci când se dorește introducerea unui element nou în baza de date, se crează mai întâi un obiect javascript pe baza unui map proprietate:valoare, iar mai apoi este trimis printr-o cerere HttpPost către un serviciu php. În scriptul php, după ce se realizează decodificarea acestuia, are loc inserarea propriu-zisă.

O problemă poate fi întâmpinată în scriptul php atunci când *magic_quotes_gpc* este activ (implicit în versiunile anterioare PHP 5.4) și se încearcă decodificarea obiectului JSON. Soluția folosită de mine a constat în apelarea metodei *stripslashes* pe elementul recepționat și utilizarea metodei de decodificare pusă la dispoziție de framework-ul php Zend.

Atunci când se dorește returnarea unor înregistrări din baza de date, procedeul este același doar că obiectul JSON este construit în scriptul PHP pe baza unei interogări a bazei de date și returnat în aplicația android sub forma unui obiect *HttpResponse* din care va fi extras stringul corespunzător obiectului JSON sau vectorului de obiecte JSON (*JSONArray*), ce vor fi ulterior decodificate și utilizate în aplicație.

JSONObject și Facebook

Întrucât modul cum aplicația interacționează cu Facebook va fi prezentat în unul dintre subcapitolele următoare, mă voi limita doar la rolul obiectelor javascript în acest proces.

¹⁸Using XML and JSON with Android, Part 1: Explore the benefits of JSON and XML in Android applications: <<[<http://www.ibm.com/developerworks/web/library/x-andbene1/index.html?ca=drs->>](http://www.ibm.com/developerworks/web/library/x-andbene1/index.html?ca=drs-)>>, accesat ultima oară la 30 mai 2012

După validarea datelor furnizate de utilizator are loc o cerere la API-ul facebook în scopul obținerii de informații cu privire la persoana logată. Acest lucru se realizează utilizând o instanță a clasei AsyncFacebookRunner, ce oferă posibilitatea executării asincrone a metodelor și returnarea rezultatelor într-o clasă ce implementează RequestListener sub forma unui string, din care se va obține în urma parsării un obiect JSON. Informațiile ce pot fi extrase se referă la numele, prenumele, emailul, numărul de telefon sau orice alte date personale memorate de Facebook.

JSONObject și Mixare

Mixare este unealta pe care am ales să o folosesc pentru a expune pe ecranul camerei date ce privesc localizarea utilizatorilor, ilustrând astfel conceptul de realitate augmentată. Mixare își procură informațiile necesare din scriptul php specificat în fișierele de configurare, iar formatul așteptat de acesta este cel corespunzător unui obiect de tip JSON. În interiorul bibliotecii, există o clasă ce are ca rol procesarea datelor extrase și transformarea lor în elemente Marker, obiecte ce vor fi afișate pe suprafața telefonului, specificând numele utilizatorului, precum și distanța la care se află de cel care face interogarea.

Facebook API

Rețeaua de socializare Facebook oferă pentru platforma Android o bibliotecă prin intermediul căreia dezvoltatorii de telefonie mobilă au posibilitatea înregistrării acesteia în aplicațiile lor. Android SDK este compus din metode ce privesc cereri sincrone (API Request Methods) și asincrone (Async API Request Methods) către serverele Facebook, metode pentru autentificare (Authentication Methods), utilizarea de ferestre Facebook (Dialog Methods) și funcții ajutătoare (Helper Methods).

Una dintre formele de logare în aplicația MeetYou constă în utilizarea contului de Facebook pentru înregistrarea și validarea datelor personale. Pentru a realiza aceste lucru, primul pas a fost în a realiza o clonă a bibliotecii Facebook¹⁹, importarea acesteia în spațiul de lucru și referențierea ei în cadrul aplicației.

¹⁹ Android SDK: <<<https://github.com/facebook/facebook-android-sdk#readme>>>, accesat ultima dată la 30 mai 2012

Următorul pas a constat în adăugarea unei semnături la proiect. Pentru aceasta, am generat o cheie cu ajutorul executabilului keytool, ce poate fi găsit în folderul Java sdk și a fișierului debug.keystore din folderul .android, urmând indicațiile prezentate pe pagina dezvoltatorilor Facebook²⁰. Șirul de caractere obținut (keyhash) a fost utilizat pentru înregistrarea aplicației MeetYou în secțiunea Mobile din App Dashboard, lucru ce poate fi vizualizat în Figura 3.3.

Apps ▶ MeetYou

[Edit App](#)[+ Create New App](#)

Settings

[Edit Settings](#)

Summary

App ID/API Key
283331348372561

App Secret
efc38c58de5d9256ff2646b3a98cd812

App Namespace
meetyouapp

Contact Email
alexandra_siriteanu@yahoo.com

Support Email
alexandra_siriteanu@yahoo.com

App Description

Figura 3.3 Înregistrarea aplicației MeetYou la Facebook

Pentru a avea acces la informațiile furnizate de rețeaua de socializare, în aplicația MeetYou a fost creat un obiect de tip *Facebook* ce a primit ca parametru identificatorul generat la înregistrare (*App ID/ App Key*) și pe baza acestuia a fost instanțiat un obiect de tip *AsyncFacebookRunner*..

Dacă proprietarul telefonului are instalat pe telefonul mobil aplicația Facebook și este logat la aceasta, MeetYou nu va trebui decât să instanțieze un obiect de tip *User* ce va prezenta datele stocate cu privire la acest utilizator, altfel o fereastră de autentificare va apărea unde utilizatorul va trebui să se logheze și să autorizeze accesul aplicației MeetYou la profilul acestuia și la graful social la care ia parte.

După autentificare, pe baza obiectului de tip *AsyncFacebookRunner* se va face o cerere la baza de date Facebook pentru obținerea datelor personale, iar acestea vor fi returnate, așa cum a fost prezentat în secțiunea anterioară sub forma unui obiect de tip JSON.

²⁰ Android Single Sign-On: <http://developers.facebook.com/docs/mobile/android/sso/#sig>, accesat ultima oară la 31 mai 2012

Mixare API

Mixare API este o bibliotecă *open-source* pentru telefoanele mobile inteligente care combină imaginea captată prin obiectivul camerei cu informațiile legate de locația dispozitivului. Atunci când telefonul este în mișcare, locația este calculată cu ajutorul metodelor prezentate anterior și apoi livrată către serviciul web specificat în fișierul de configurare. Punctele de interes aflate în apropiere sunt serializate sub forma unui obiect JSON și returnate aplicației. Informațiile ce sunt vizibile pe ecranul telefonului depind de direcția în care este îndreptat dispozitivul camerei, aspect detectat cu ajutorul compasului, giroscopului și a senzorilor de mișcare.

Programatorul are posibilitatea specificării mai multor servicii web, de la care se așteaptă răspuns și codificarea rezultatelor folosind diverși markeri, în funcție de serviciul de care aparțin. Astfel, există posibilitatea utilizării formelor preexistente: cercuri, triunghiuri ce pot lua culori diferite, dar se pot introduce și alte forme sau elemente de tip *Drawable*.

Obiectul JSON recepționat în urma apelării la serviciul web este mai apoi procesat pe baza structurii așteptate și transformat într-un vector de obiecte *POIMarker* ce va fi afișat pe ecranul telefonului. Fiecare obiect *POIMarker* este constituit dintr-o iconiță și o casetă text, în care este specificat punctul de interes și distanță la care se află de utilizator.

Pentru a folosi aceste funcționalități oferite de Mixare, a fost necesar clonarea aplicației Mixare Browser aflată la adresa <https://github.com/mixare/mixare.git> și împărțirea pachetelor în funcție de rolul pe care îl au în crearea spațiului augmentat. Pe de o parte a fost necesar crearea și importarea unui proiect bibliotecă unde au fost puse pachetele de forma *org.mixare.lib*, iar pe de altă parte au trebuit adăugate în cadrul proiectului MeetYou celelalte pachete ce conțineau clasele responsabile cu accesarea serviciilor web și procesarea informațiilor pentru expunerea lor pe ecranul telefonului.

În cadrul aplicației MeetYou, biblioteca Mixare a fost utilizată pentru afișarea prietenilor pe dispozitivul camerei. Acest lucru a fost realizat prin crearea unui serviciu web care pe baza identicatorului utilizatorului returnează locația prietenilor (stocată în baza de date), folosind următorul format:


```
{
    "status": "OK",
    "num_results": 1,
    "results": [
        {
            "id": "2",
            "lat": 47.173752,
            "lng": "27.575133",
            "elevation": "0",
            "title": "Ion Tudorache",
            "distance": "380",
            "has_detail_page": "0",
            "webpage": ""
        }
    ]
}
```

Adresa serviciului a fost specificată în lista de șiruri de caractere „defaultdatasource” din fișierul `arrays.xml`, odată cu aceasta și numele serviciului – *MEETYOU* – precum și tipul de iconiță folosit – *NAVIGATION_MARKER*. În cazul în care un prieten se află la o distanță mai mică de 100 de metri, iconița va fi constituită dintr-un triunghi verde, altfel va fi o resursă de tip *Drawable*.

Mesaje Text (Short Message Service)

Una dintre funcționalitățile utile pe care Android o pune la dispoziția programatorilor este posibilitatea trimiterii de mesaje în interiorul aplicațiilor, precum și recepționarea și manipularea conținutului acestora atunci când sunt recepționate de către dispozitivul mobil.

Trimiterea de mesaje text

Clasa *SmsManager* face parte din pachetul *android.telephony* și este responsabilă cu trimiterea de date, text și mesaje pdu²¹ SMS. Pentru a putea trimite mesaje, este necesar obținerea unei referințe la un obiect *SmsManager* utilizându-se metoda statică *SmsManager.getDefault*, precum și specificarea în fișierul manifest al proiectului permisiunea `<uses-permission android:name="android.permission.SEND_SMS"/>`. Ultimul pas constă în apelarea metodei *sendTextMessage* ce are printre parametri numărul destinatarului și mesajul ce se dorește a fi livrat.

²¹ Protocol data unit

Recepționarea de mesaje text

Pentru ca o aplicație să urmărească difuzarea de intenții SMS, este necesară specificarea permisiunii `<uses-permission android:name="android.permission.RECEIVE_SMS"/>` în manifestul aplicației, precum și înregistrarea unui *Broadcast Receiver* ce va declanșa o nouă intenție de difuzare (*Broadcast Intent*) cu acțiunea `android.provider.Telephony.SMS_RECEIVED` de fiecare dată când un mesaj este recepționat.

Intenția de difuzarea a mesajelor include informații privind mesajele text recepționate (numărul de telefon al destinatarului, timpul la care a avut loc livrarea), salvate sub forma unui vector de obiecte cu formatul PDU²². Pentru extragerea acestor informații este necesară transformarea componentelor vectorului byte PDU în obiecte *SmsMessage* prin apelarea metodei *SmsMessage.createFromPdu*, așa cum este ilustrat în următoarea secvență.

```
Bundle bundle = intent.getExtras();
SmsMessage[] msgs = null;
if (bundle != null) {
    Object[] pdus = (Object[]) bundle.get("pdus");
    msgs = new SmsMessage[pdus.length];
    for (int i=0; i<msgs.length; i++){
        msgs[i] = SmsMessage.createFromPdu((byte[])pdus[i]);
    }
}
```

Nivelul de interfață

Interfața unei aplicații este tot ceea ce utilizatorul poate vedea și cu care poate interacționa. Platforma Android oferă o varietate de componente preconstruite cum ar fi obiectele de aspect²³ și controalele UI ce permit modelarea interfeței grafice Android.

Toate elementele de interfață într-o aplicație Android sunt construite utilizând obiectele *View* și *ViewGroup*. Un obiect de tip *View* este un element care desenează ceva pe ecran și care permite interacțiunea cu utilizatorul. Un obiect *ViewGroup* este un container care adăpostește mai multe

²² Protocol Data Unit

²³ Layouts, în limba engleză

obiecte View pentru a putea defini aspectul unei interfețe, fiecare componentă a aplicației fiind definită ca o ierarhie de obiecte View și ViewGroup, așa cum poate fi vizibil în Figura 4.1.

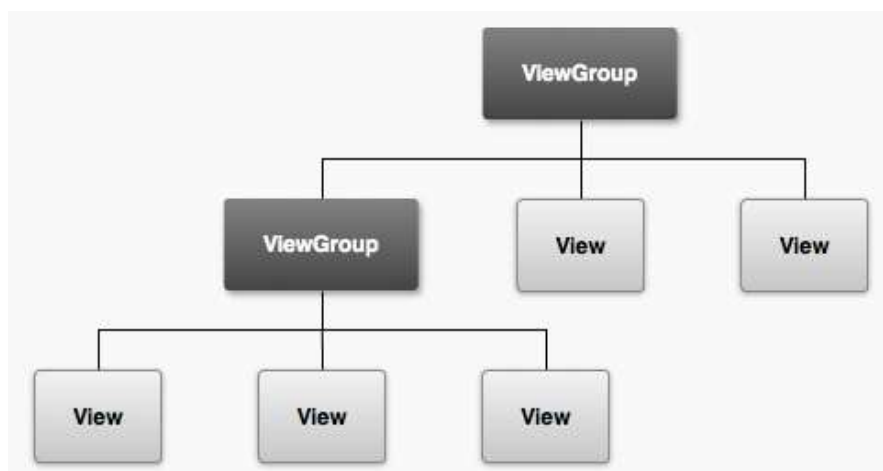


Figura 3.4 Ierarhie de obiecte View și ViewGroup

În cadrul acestei secțiuni nu mă voi concentra asupra fiecărui element de vizualizare utilizat în realizarea acestei interfețe grafice, ci voi încerca să explic elementele avansate pe care le-am utilizat în crearea ei.

Lucrul cu animații

Interfețele grafice pentru aplicațiile mobile s-au dezvoltat într-un ritm alert, permițând folosirea unor elemente avansate de grafică cum sunt umbrele, efectele de transluciditate, animațiile sau utilizarea bibliotecii OpenGL.

Animațiile Tweened oferă o modalitate simplă de a oferi adâncime, mișcare sau feedback utilizatorilor la un cost minim de resurse. Utilizarea de animații pentru a aplica un set de schimbări în privința orientării, scalării, poziționării și opacizării resurselor este mai puțin costisitoare decât redesenarea manuală a spațiului de lucru pentru a obține efecte similare.

Animațiile Tweened sunt adesea folosite pentru:

- Tranziția dintre activități

- Tranziția între aspectele²⁴ unei activități
- Tranziția dintre diferite părți ale conținutului afișat în același ecran
- Oferirea unui feedback vizual utilizatorului (e.g. o clepsidră care se rotește indicând progresul unei acțiuni)

Crearea de animații Tweened

Animațiile Tweened sunt create utilizându-se clasa *Animation* din pachetul *android.view.animation*. Următoarea listă oferă o imagine asupra tipurilor de animații disponibile:

- *AlphaAnimation* permite animarea unei schimbări în perspectiva vizuală în ceea ce privește transparența
- *RotateAnimation* permite rotirea elementului de vizualizare selectat în planul XY
- *ScaleAnimation* oferă posibilitatea măririi sau micșorării obiectului selectat
- *TranslateAnimation* se utilizează pentru mutarea elementului selectat pe suprafața ecranului

Clasa *AnimationListener* permite crearea unei rutine de tratare a unui eveniment care este declanșat atunci când o animație începe sau se termină. Acest lucru se poate realiza apelând metoda *setAnimationListener* pe obiectul *Animation* și plasarea unei implementări pentru interfața *AnimationListener*, suprascriind metodele *onAnimationEnd*, *onAnimationStart* și *onAnimationRepeat* după necesități.

Totodată poate fi precizat modul cum o animație variază în timp prin intermediul unui obiect de tip *Interpolator*. Acest lucru permite ca efectele de animație (transparență, scalare, translare, rotație) să fie accelerate, încetinite sau repetate.

În cadrul aplicației MeetYou, animațiile au fost folosite pentru translarea unor obiecte de tip *View* atunci când un buton este apăsător. Mai exact, atunci când butonul de opțiuni este atins, obiectul de tip *LinearLayout* în care sunt definite elementele din meniu va deveni vizibil și va transla spre partea opusă a ecranului. La o doua atingere, acesta va transla spre stânga și va deveni din nou

²⁴ layouts, în limba engleză

invizibil, lucru ilustrat în Figura 3.5. Menționez că la realizarea acestei funcționalități am folosit ca sursă de inspirație postarea de pe blogul Android Blogger²⁵.

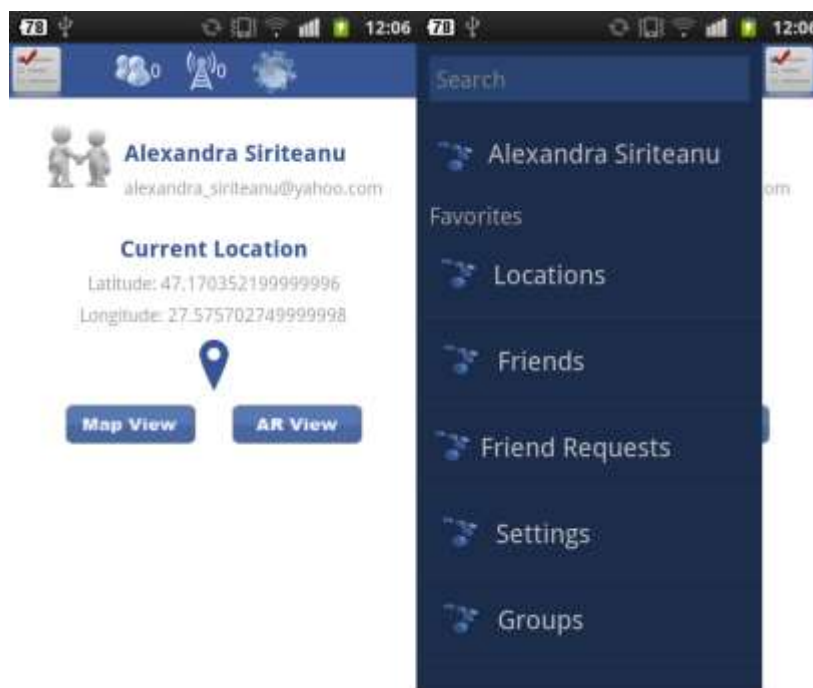


Figura 3.5 Translarea unui obiect de tip `LinearLayout`

Utilizarea de liste cu conținut eterogen

Obiectele `ListView` puse la dispoziție de către Android sunt un mod excelent de a expune date care au aceeași configurație. Cu toate acestea, de cele mai multe ori conținutul unei liste variază și este nevoie de o structură mai complexă pentru a expune aceste informații.

În cazul aplicației *MeetYou*, listele eterogene au fost utilizate pentru a grupa prietenii unui utilizator în funcție de grupul de care aparțin. Pentru aceasta m-am folosit de postarea *CWAC'd Up*:

²⁵Android Blogger, Sliding drawer, again aflat la adresa << <http://androidblogger.blogspot.ro/2009/01/sliding-drawer-again.html>>>, accesat ultima oară la 20 iunie 2012.

Alternative Adapters de pe blogul adresat dezvoltatorilor de aplicații pentru telefoane mobile: AndroidGuys²⁶.

Pentru a utiliza această facilitate a fost necesară clonarea proiectului de la adresa <https://github.com/commonsguy/cwac-merge/tree>, importarea bibliotecii java *SackOfViewsAdapter.jar* și adăugarea în cadrul proiectului a clasei *MergeAdapter*. Pentru a crea o listă cu elemente având o structură distinctă a fost necesară crearea unui obiect de tip *MergeAdapter* și adăugarea unui obiect de tip *ListAdapter* pentru fiecare tip de listă utilizat și a un obiect de de tip *View* pentru fiecare element de vizualizare introdus.

Nivelul de date

Informațiile despre utilizatorii aplicației MeetYou sunt stocate într-o bază de date MySQL, a cărei schemă este prezentată în Figura 3.6.

Atunci când un utilizator dorește să-și creeze un cont nou, informațiile colectate vor fi stocate în tabelul *Potential_Users*, iar mai apoi după ce utilizatorul va confirma cererea făcută, informațiile vor fi adăugate în tabelul *Users*. În cazul în care un utilizator dorește să se logheze cu contul de Facebook, informațiile vor fi stocate direct în tabelul *Users*.

Atunci când un utilizator dorește să adauge un alt utilizator în lista de prieteni, acesta va trimite o cerere și o noua înregistrare va fi făcută în tabelul *New_Requests*. După ce al doilea utilizator va accesa notificarea, datele vor fi adăugate în tabelul *Requests*. În cazul în care acesta dorește să accepte prietenie, se vor adăuga două linii în tabelul *Friends*, corespunzătoare fiecăruia dintre utilizatori, altfel se va adăuga o linie în tabelul *Rejected_Requests* care va împiedica o alta tentativă de a primi o cerere din partea acelui utilizator. Fiecare utilizator are posibilitatea adăugării de noi grupuri și locații ce vor fi stocate în tabelele *Groups*, respectiv *Saved_Locations*. Fiecare grup poate avea setată o anumită locație, astfel încât dacă utilizatorul se află în perimetrul acelei locații și alți prieteni se află în apropierea lui, acesta nu va primi notificare decât de la cei care nu se află în acel grup. Locațiile salvate vor fi accesate și atunci când un utilizator dorește să se înregistreze într-o

²⁶ Mark Murphy, CWAC'd Up: Alternative Adapters, Android Guys, în format electronic la adresa <<<http://www.androidguys.com/2009/07/29/cwacd-up-alternative-adapters/>

anumită locație, lucru ce va fi salvat în tabelul *Checked_In* și va fi afișat pe peretele de noutăți al prietenilor săi.

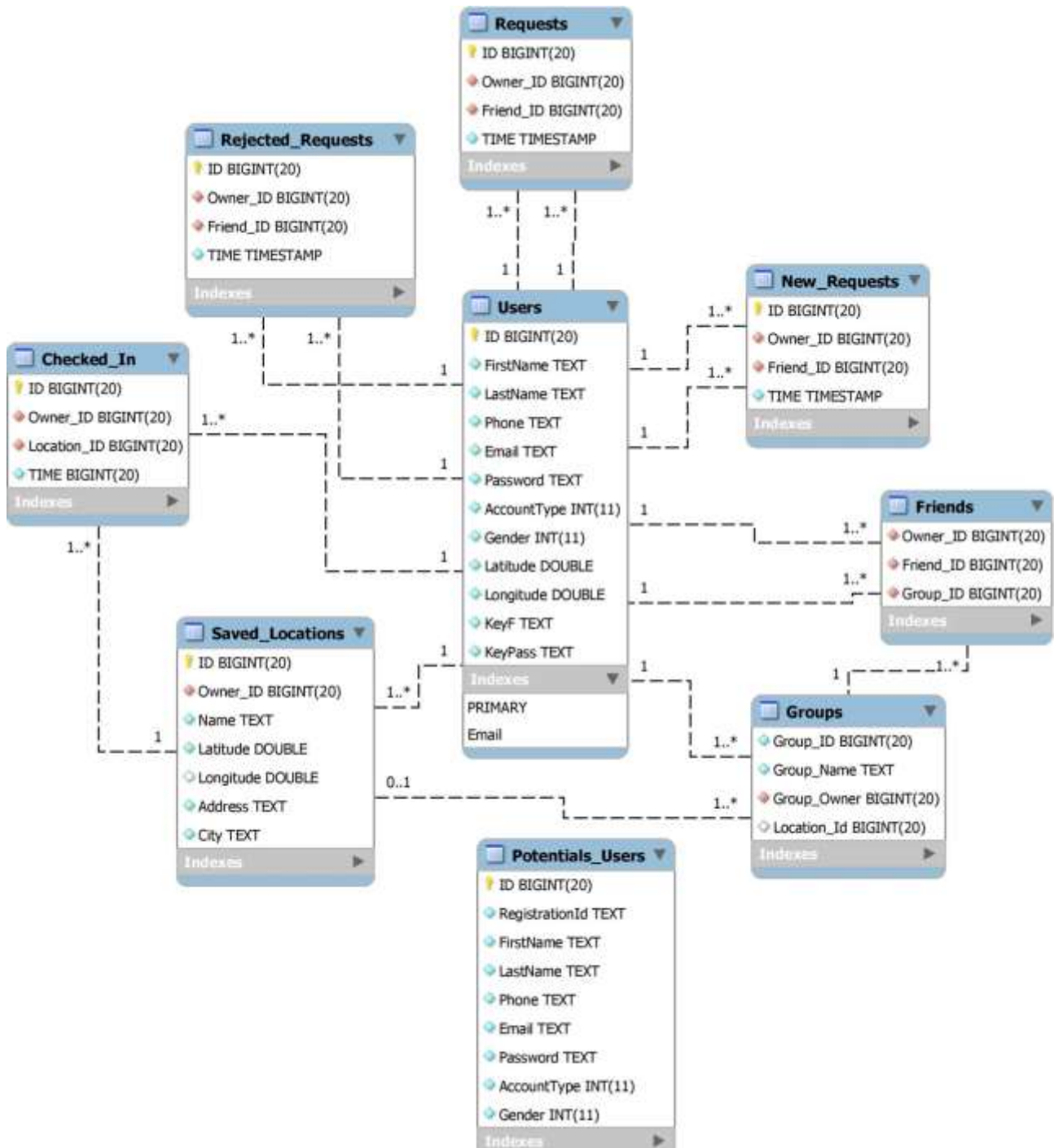


Figura 3.6 Diagrama bazei de date

Ghid de utilizare a aplicației

Crearea unui cont nou

Pentru a putea utiliza această aplicație, o persoană are două posibilități:

- Crearea unui nou cont, utilizând opțiunea *Get a new MeetYou ID!*
- Crearea unui nou cont, utilizând contul de Facebook

Crearea unui cont MeetYou constă în doi pași:

- completarea unor informații personale referitoare la numele, prenumele, sexul, numărul de telefon și emailul persoanei, precum și parola ce urmează a fi utilizată pentru logare (Figura 4.1)
- activarea contului, accesând adresa primită prin email

Crearea unui cont utilizând autentificarea Facebook se va putea realiza la prima logare utilizând butonul *Connect*, existând două posibilități:

- dacă utilizatorul are aplicația de Facebook instalată și este deja logat, informațiile extrase din contul de Facebook vor fi folosite pentru crearea contului MeetYou, după aceasta urmând logarea propriu-zisă în aplicație
- dacă aplicația Facebook nu este instalată pe dispozitivul Android, o fereastră Facebook de logare va apărea pe ecran, unde acesta își va putea introduce datele personale. Dacă acestea sunt corecte, va fi creat un nou cont MeetYou și va putea începe să utilizeze aplicația

Logarea în aplicația MeetYou

Pentru ca un utilizator să se poată loga cu contul MeetYou, acesta trebuie să completeze câmpurile corespunzătoare adresei de email și parolei pe care și-a ales-o la crearea contului și să apese butonul *Sign in*. Dacă informațiile sunt corecte, acesta va fi direcționat către profilul său. Acesta are posibilitatea memorării datelor, bifând opțiunea *Remember my ID and password*, astfel încât la următoarea logare nu va mai fi nevoie să completeze câmpurile.

De fiecare dată când cineva se va loga în aplicație utilizând contul de Facebook, se va realiza o cerere asincronă la serviciul Facebook pentru obținerea informațiilor personale, iar datele obținute vor fi utilizate la actualizarea bazei de date MeetYou.

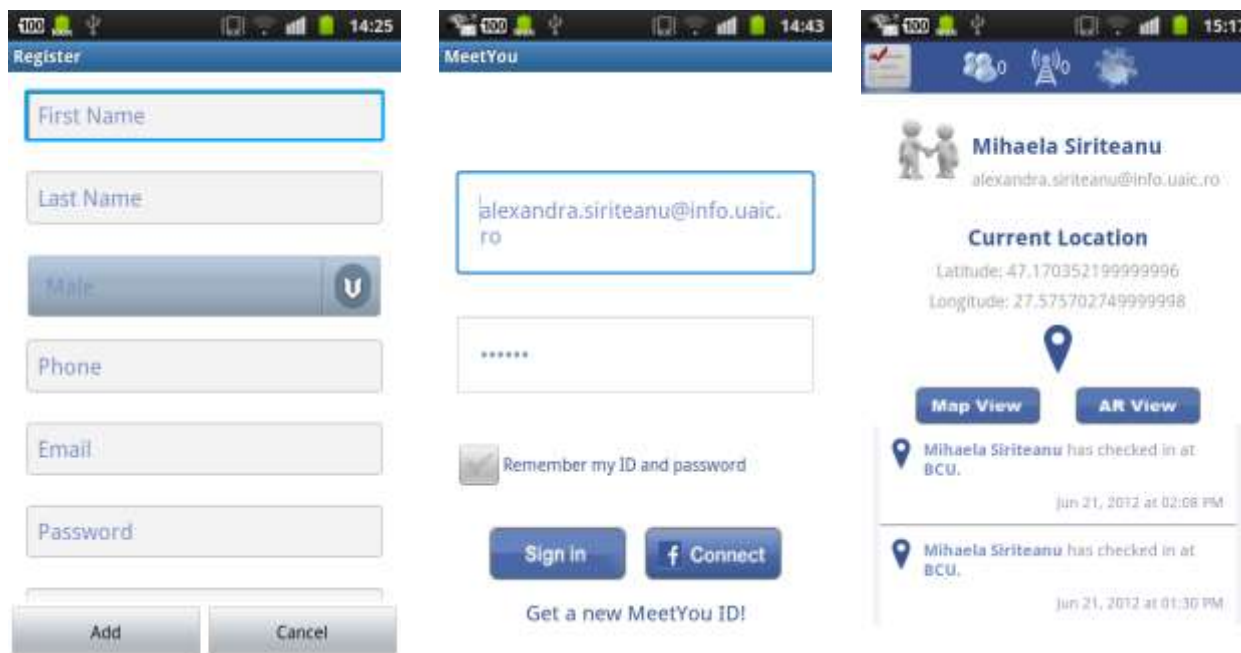


Figura 3.7 Crearea unui cont MeetYou și logarea în aplicație

Modelul aplicației

Aplicația este formată dintr-un cadru principal unde fiecare dintre opțiunile selectate va fi încărcată, dintr-o bară de instrumente compusă din trei butoane corespunzătoare accesării cererilor de prietenie, noutăților și vizualizării locațiilor salvate, precum și dintr-un buton corespunzător accesării meniului lateral.

Meniul lateral este format dintr-o casetă text unde pot fi căutați alți utilizatori, o opțiune către pagina principală și o listă cu toate funcțiile pe care le posedă aplicația: vizualizarea grupurilor, locațiilor, noutăților, setărilor și a utilizatorilor respinși.

Profilul utilizatorului

Pagina principală a aplicației constă în expunerea informațiilor principale ale utilizatorului: nume, prenume, email, locația curentă dată prin latitudine și longitudine, precum și lista cu ultimele zece înregistrări ale locației. Pe această pagină se găsește butonul de *Check In* (pentru înregistrarea într-o anumită locație), precum și două butoane de vizualizare a prietenilor. Apăsându-se pe butonul *Map View*, informațiile referitoare la localizarea prietenilor vor fi expuse pe suprafața hărților Google, permițându-i utilizatorului să aibă o privire de ansamblu asupra tuturor locațiilor. Butonul *AR View* va deschide camera telefonului mobil și va afișa pe suprafața ecranului prietenii aflați la o distanță de maxim douăzeci de kilometri.

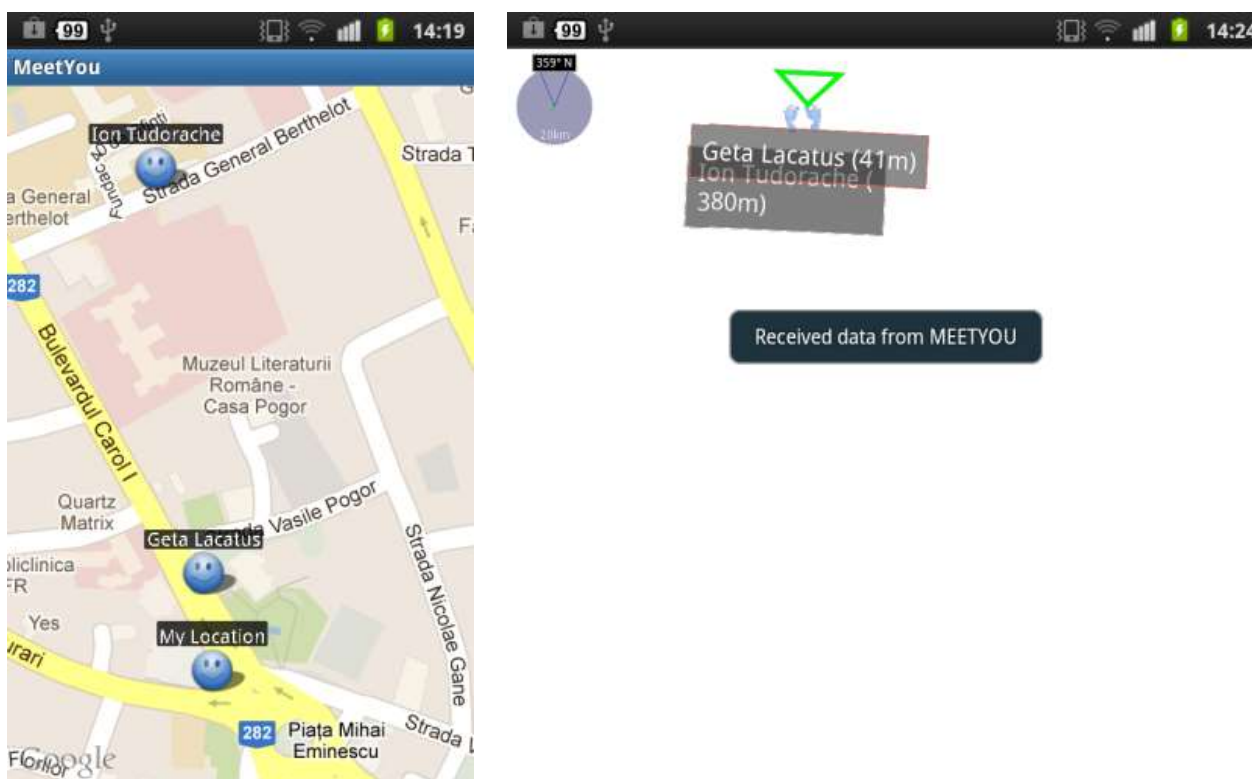


Figura 3.8 Afișarea prietenilor pe hărțile Google și pe ecranul telefonului utilizând Mixare

Adăugarea, editarea, ștergerea și vizualizarea unei locații

Utilizatorul are posibilitatea să se înregistreze în anumite locații pentru ca prietenii săi să fie înștiințați în privința locației și opțional asupra activității pe care o desfășoară. Pentru aceasta, utilizatorul trebuie să își salveze propriile locații, urmând următorii pași:

- Alegerea opțiunii *Locations* din meniul cu opțiuni sau din bara cu instrumente

- Apăsarea butonului *plus* din caseta text *Search Locations*
- Completarea câmpurilor cu informațiile necesare: numele, latitudinea, longitudinea, orașul și adresa locației
- Apăsarea butonului de *Save*

Pentru a vizualiza locația se va apăsa pe butonul desemnat de imaginea globului pământesc, existând posibilitatea schimbării latitudinii și longitudinii odată ce pinul este mutat în altă zonă a hărții. Câmpurile ce trebuiesc completate cu adresa, respectiv orașul corespunzător locației vor fi setate în mod implicit cu informațiile obținute utilizând serviciul Google Maps.



Figura 3.9 Crearea și vizualizarea unei locații utilizând Google Maps

În cazul în care se dorește editarea sau ștergerea unui locații, aceasta va fi mai întâi căutată utilizând căseta text *Search Locations*, iar în urma efectuării unei apăsări prelungite se va putea alege opțiunea dorită din meniul de context apărut.

Opțiunea Check In

Pentru a se putea înregistra într-o anumită locație, utilizatorul va trebui să apese butonul imagine de pe pagina principală a profilului. O listă cu toate locațiile salvate va deveni accesibilă, iar în urma selectării uneia dintre acestea și apăsării butonului *Add*, o fereastră dialog va apărea în care persoana va putea să precizeze activitatea pe care o desfășoară. Apăsând butonul OK, înregistrarea

va fi efectuată, iar prietenii vor putea avea acces la această modificare, accesând profilul acestui utilizator sau vizualizând lista cu noutăți.



Figura 3.10 Înregistrarea într-o locație

Vizualizarea noutăților

Al doilea buton din bara de instrumente are ca scop afișarea locurilor unde s-au înregistrat prietenii utilizatorilor, lista fiind actualizată cu date mai vechi pe măsură ce persoana defilează printre elementele acesteia.

Noi cereri de prietenie

Atunci când utilizatorul primește noi cereri de prietenie, acesta va fi înștiințat prin afișarea numărului de cereri ce nu au fost vizualizate pe suprafața butonului din bara de instrumente. Atunci când acesta este apăsât aceste cereri vor putea fi vizualizate și se va putea răspunde prin acceptul sau respingerea acestora. În cazul în care utilizatorul nu dorește să răspundă la acestea, cererile vor putea fi vizualizate mai târziu accesând opțiunea *Friend Requests* din meniul lateral.

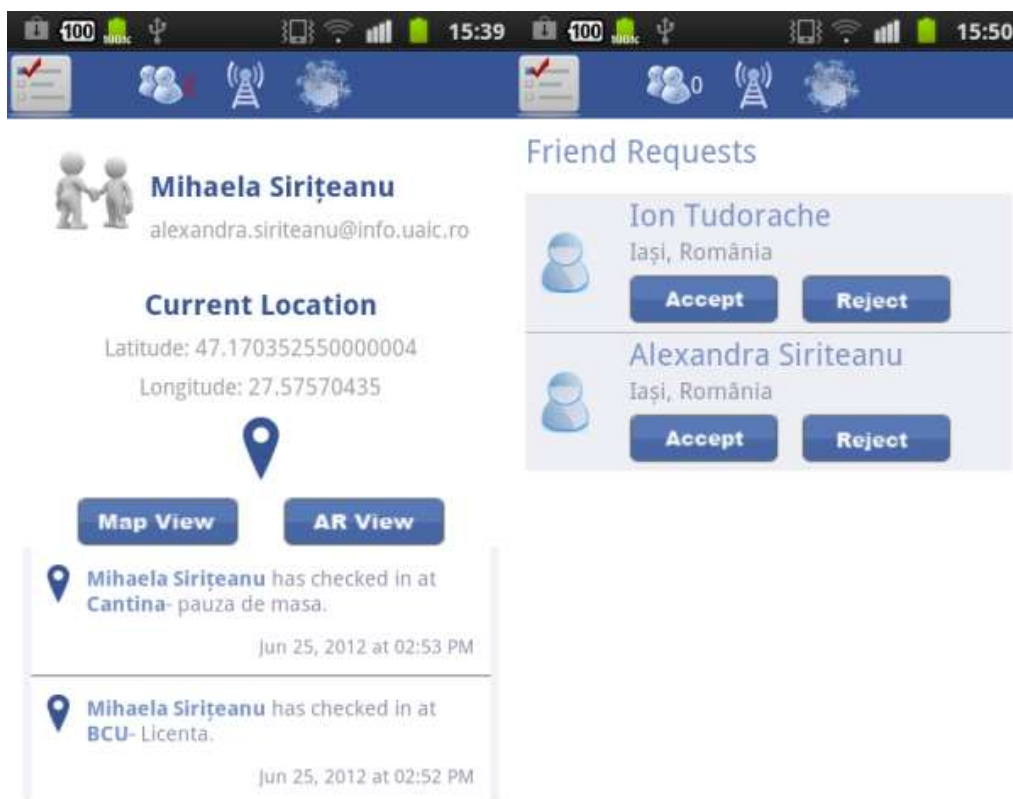


Figura 3.11 Recepționarea de cereri noi și afișarea lor

Administrarea grupurilor

Un utilizator are posibilitatea definirii mai multor grupuri cu ajutorul cărora să organizeze lista de prieteni. Fiecare grup poate avea asociat o locație, astfel încât atunci când se află în preajma acelei locații (la o distanță de maxim 200 de metri), alarma care notifică faptul că se află în preajma unor cunoștințe să nu se declanșeze decât pentru persoanele care nu se află în acest grup. Acest lucru este necesar atunci când, spre exemplu, utilizatorul se află în cadrul facultății în care studiază și nu ar vrea să fie notificat de prezența colegilor, dar ar vrea să fie notificat în cazul în care un alt prieten ar fi în apropiere.

Atunci când utilizatorul trimite sau acceptă o cererea de prietenie trebuie să specifice grupul în care va dori să adauge acea persoană, nefiind posibilă existența unui prieten în afara unui grup. Aplicația permite și editarea componentei grupurilor, utilizatorul putând schimba grupul din care face parte un utilizator printr-o simplă apăsare de buton de pe pagina de profil a celui prieten.

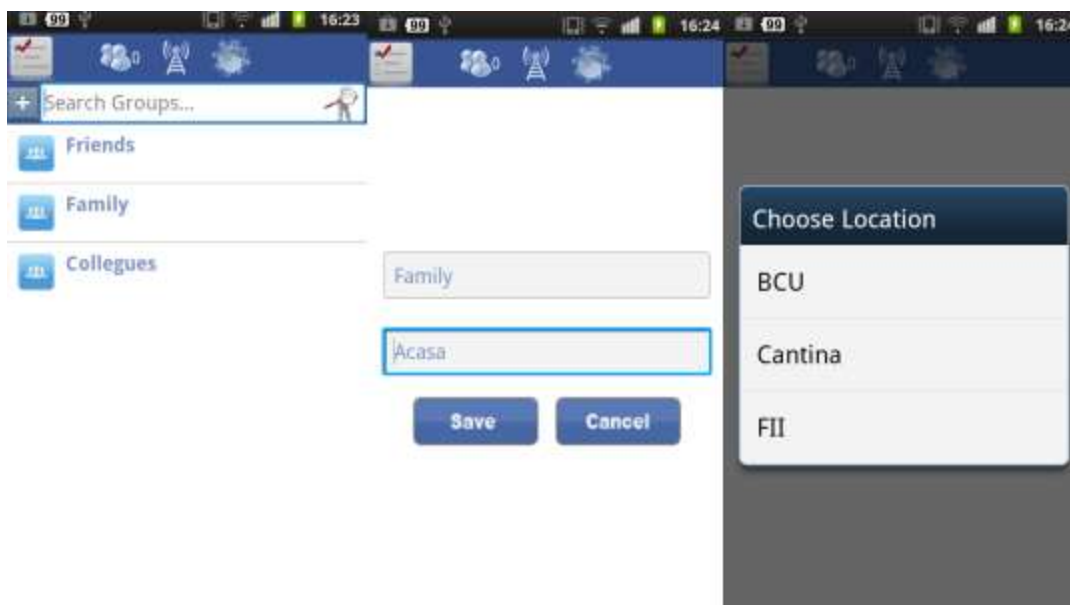


Figura 3.12 Editarea unui grup

Acceptarea utilizatorilor a căror cerere a fost respinsă

Atunci când un utilizator respinge cererea de prietenie a unui alt utilizator, persoana respinsă nu va mai putea să-și reînoiească cererea. Pentru aceasta, în cazul în care primul utilizator se răzgândește va putea să accepte mai târziu prietenia prin deschiderea listei *Rejected Users*, găsirea utilizatorului dorit, accesarea profilului acestuia și apăsării butonului de acceptare.



Figura 3.13 Adăugarea unui utilizator a cărei cerere de prietenie a fost respinsă

Vizualizarea prietenilor

Pe lângă cele două metode prezentate mai sus de afișare a prietenilor, o altă cale este accesând opțiunea *Friends* din meniul lateral. O listă cu toți prietenii așezați în funcție de grupul de care aparțin va deveni accesibilă, iar prin selectarea unuia dintre prieteni va putea fi vizualizat profilul acestuia.

Pagina de profil a acestuia pune la dispoziție posibilitatea schimbării grupului de care aparține, ștergerea acestuia ca prieten, accesarea drumului rutier între utilizator și acesta, precum și lista cu prietenii acestuia.



Figura 3.14 Accesarea opțiunilor disponibile pe profilul unui prieten

Trimiterea locației pe baza unui cuvânt cheie

Utilizatorul are posibilitatea alegerii unui cuvânt cheie pe baza căruia o altă persoană care cunoaște această cheie să poată accesa informațiile privind locația acestuia prin trimiterea unui mesaj cu textul constituit din acest cuvânt. Pentru schimbarea cuvântului cheie, utilizatorul va trebui să-și stabilească o parolă și să acceseze opțiunea *Settings* din meniu. Inițial atât cheia cât și parola coincid, fiind reprezentate de șirul de caractere „MeetYou”.

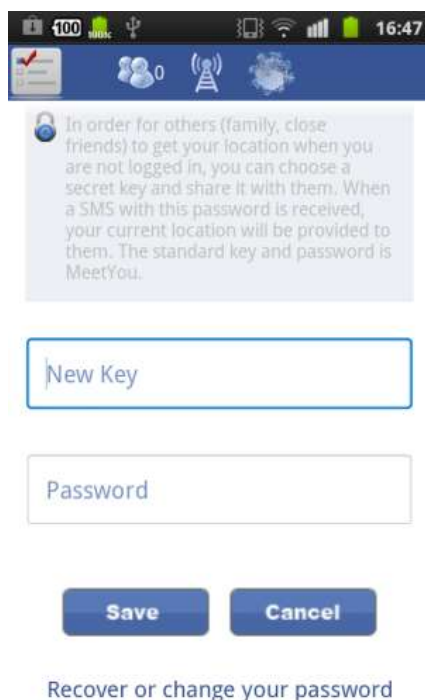


Figura 3.15 Setarea cuvântului cheie

Căutarea unui prieten

Apăsând caseta text *Search* se va deschide o nouă fereastră unde în urma tastării numelui dorit vor fi filtrați prietenii și prietenii prietenilor a căror nume începe cu șirul de caractere tastat. În cazul în care nu am găsit persoana căutată se va putea apăsa butonul *Get more results*, care va returna o listă cu toți utilizatorii ce folosesc această aplicație și au un nume care începe cu textul tastat.



Figura 3.16 Căutarea unui prieten

Prin accesarea unuia din rezultate se va putea vizualiza profilul acelui utilizator. În cazul în care acesta este prieten cu cel care face căutarea, toate informațiile vor fi afișate, altfel un mesaj corespunzător cu relația pe care aceștia o au va apărea. Mesajul poate fi unul referitor la necesitatea de a adăuga acel utilizator în lista de prieteni, de a aștepta confirmarea cererii făcute, de accepta cererea primită pentru a putea avea acces la datele acestui utilizator.

Rezumat

Aplicația MeetYou este alcătuită din trei module corespunzătoare celor trei nivele ale modelului arhitectural client – server Three-tier (3-tier architecture), unde elementele de interfață, de procesare și datele sunt separate. Nivelul de prezentare este constituit de aplicația Android cu ajutorul căreia vor fi expuse informațiile referitoare la utilizatori, nivelul de date este reprezentat de baza de date în care sunt stocate aceste informații, iar nivelul logic reprezintă legătura dintre cele două nivele și este realizată cu ajutorul serviciilor web și a serviciilor Android.

Rolul acestei aplicații este de a crea o rețea socială ce permite schimbul de informații privind locația utilizatorilor și activitățile pe care le desfășoară. Pentru aceasta, au fost utilizate serviciile de localizare puse la dispoziție de Android, care cu ajutorul tehnologiei GPS, a rețelei wireless și a turnurilor de telefonie mobilă reușește să determine poziția aproximativă a telefonului mobil pe care rulează acest program. Afișarea acestor informații se realizează utilizând hărțile Google, listele eterogene CWAC și realitatea augmentată obținută prin utilizarea bibliotecii Mixare.

Totodată se oferă posibilitatea utilizatorilor de a fi notificați atunci când sunt în apropierea unor prieteni și de a seta locațiile în care nu doresc să fie notificați în privința unor grupuri de persoane (e.g. notificările apropierii de anumiți colegi atunci când utilizatorul este la facultate).

O altă caracteristică importantă este posibilitatea partajării locației în momentul când utilizatorul nu are deschisă aplicația. Prin setarea unui cuvânt cheie, alte persoane ce se află în posesia cheii pot afla locația curentă a utilizatorului prin trimiterea unui mesaj format din cuvântul selectat de acesta.

Concluziile lucrării

Lucrarea de față are ca scop prezentarea unei aplicații care să ilustreze caracterul social al omului - nevoia de a fi mereu în legătură cu familia și prietenii săi – avându-se în vedere facilitățile puse la dispoziție de sistemul de operare Android.

Din punct de vedere structural, o aplicație Android este compusă din trei nivele principale: partea de interfață pe care o persoană o poate vedea și cu care poate interacționa, partea logică care obține informațiile cerute de utilizator și partea de date cu ajutorul căreia sunt salvate informațiile necesare logicii aplicației.

Rolul acestui program este de a crea o rețea socială ce permite schimbul de informații privind locația utilizatorilor și activitățile pe care le desfășoară. Pentru aceasta, au fost utilizate serviciile de localizare puse la dispoziție de Android, care cu ajutorul tehnologiei GPS, a rețelei wireless și a turnurilor de telefonie mobilă reușește să determine poziția aproximativă a telefonului mobil pe care rulează acest program. Afișarea acestor informații s-a realizat utilizând atât componentele preconstruite puse la dispoziție de platforma Android, cât și elemente mai complexe ce includ listele eterogene CWAC, hărțile Google și realitatea augmentată obținută prin utilizarea bibliotecii Mixare.

Noutatea pe care această aplicație o oferă se referă la modul cum sunt îmbinate facilitățile oferite de platforma Android. În domeniul aplicațiilor de telefonie mobilă există deja programe sociale care folosind serviciul de localizare oferă informații utilizatorului despre localizarea prietenilor săi sau programe care utilizând tehnologia realității augmentate oferă informații despre obiectivele aflate în apropierea acestuia, în schimb nu există nicio aplicație care să le îmbine pe amandouă oferind o rețea socială ce utilizează realitatea augmentată. Totodată există aplicații care se folosesc de notificări pentru a-l atenționa pe utilizator de existența unor evenimente, dar în aplicațiile sociale studiate de mine nu am găsit nicio aplicație care să atenționeze doi prieteni atunci când se găsesc în apropiere, oferindu-le posibilitatea de a se întâlni. Pentru că este cunoscut consumul mare de energie pe care o aplicație îl poate avea atunci când este conectată la internet, programul realizat de mine oferă și posibilitatea de a trimite locația printr-un mesaj electronic la recepționarea unui SMS

format dintr-un cuvânt cheie (ales de utilizator de partajat cu persoanele dorite) chiar și atunci când utilizatorul nu utilizează această program.

În viitor, aplicația ar putea fi îmbunătățită prin adăugarea unor elemente mai complexe de interfață, prin utilizarea unor biblioteci ce va face posibilă recunoașterea de comenzi vocale sau prin utilizarea datelor oferite de senzorii de mișcare pentru a detecta anumite semnale în scopul deschiderii și utilizării anumitor opțiuni din aplicație. Totodată ar putea fi implementate facilități precum: adăugarea unor imagini pentru profilul utilizatorului, partajarea de poze luate la locul unde utilizatorul s-a înregistrat și posibilitatea realizării unui spațiu de conversație unde utilizatorii ar putea comunica în scopul stabilirii unei întâlniri.

În concluzie, consider că platforma Android pune la dispoziție instrumentele necesare pentru crearea unor aplicații care să ajute la o mai buna interrelaționare a oamenilor, oferind elementele necesare pentru realizarea unui interfețe care să atragă utilizatorii și cu care aceștia să poată ușor interacționa. Ceea ce va face diferența între aplicații va fi modul cum aceste componente vor fi îmbinate și utilizate pentru îndeplinirea scopului propus.

Bibliografie

[1] Aristotel, Politica I.

[2] Speckmann, B. 2008. The Android mobile platform. A Review Paper Submitted to the Eastern Michigan University Department of Computer Science In Partial Fulfillment of the Requirements for the Master of Science in Computer Science. Ypsilanti, Michigan. (în format electronic la: <<http://www.emich.edu/compsci/projects/Master_Thesis_-_Benjamin_Speckmann.pdf>>, accesat ultima oară la 17 decembrie 2011).

[3] Pettey, C., Goasduff, L., Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent, Ziarul Gartner, Egham, UK, August 11, 2011 (în format electronic la: <<<http://www.gartner.com/it/page.jsp?id=1764714>>>, accesat ultima oară la 17 decembrie 2011)

[4] Android (Sistem de operare), Wikipedia. (în format electronic la: <<[http://ro.wikipedia.org/wiki/Android_\(sistem_de_operare\)](http://ro.wikipedia.org/wiki/Android_(sistem_de_operare))>>, accesat ultima oară la 17 decembrie 2011).

[5] <http://www.citatepedia.ro/citat.php?id=40965>, accesat ultima oară la 17 decembrie 2011

[6] Hoog, A., 2011, Android Forensics Investigation, Analysis and Mobile Security for Google Android, Elsevier, Inc., Capitolele 2-3

[7] Android Developers: <<<http://developer.android.com/guide/basics/what-is-android.html>>>, accesat ultima oară la 18 decembrie 2011.

[8] Dube, R., 2010, 5 Mobile Social Networks That Know When Your Friend Is Near, (în format electronic la: <<<http://www.makeuseof.com/tag/mobile-social-networks-friend/>>>, accesat ultima oară la 19 decembrie 2011)

[9] Pagina oficială Latitude: <<<http://www.google.com/mobile/latitude/>>>, accesată ultima oară la

19 decembrie 2011

[10] Imagine ilustrativă Gypsii: <<<http://www.gypsii.com/images/gypsii.jpg>>>, accesată ultima oară la 3 ianuarie 2012

[11] Video demonstrativ Gypsii: <<<http://www.youtube.com/watch?v=uATrnzGQCx8>>> accesat ultima oară la 19 decembrie 2011

[12] Pagina oficială LociMobile: <<<http://www.locimobile.com/apps/>>>, accesată ultima oară la 19 decembrie 2011

[13] Pagină oficială Look!: <<<http://www.lookar.net>>>, accesată ultima oară la 28 decembrie 2011

[14] Metaio Mobile SDK Documentation, (în format electronic la: <<<http://docs.metaio.com/bin/view/Main/UnifeyeMobile>>>, accesat ultima oară la 28 decembrie 2011)

[15] Pagina oficială Junaio: <<<http://www.junaio.com/>>>, accesată ultima oară la 28 decembrie 2011

[16] Pagina oficială Mixare: <<<http://www.mixare.org/>>>, accesată ultima oară la 28 decembrie 2011

[17] Mixare – A New Augmented Reality Engine For Android, (în format electronic la: <<<http://www.augmentedplanet.com/2010/03/mixare-a-new-augmented-reality-engine-for-android/>>>, accesat ultima oară la 28 decembrie 2011)

[18] Pagina oficială Wikitude SDK <<<http://www.wikitude.com/developer/documentation/wikitude-sdk>>>, accesat ultima oară la 28 decembrie 2011

[19] Butchart, B., Augmented Reality for Smartphones, (în format electronic la: <<http://observatory.jisc.ac.uk/docs/AR_Smartphones.pdf>>, accesat ultima oară la 19 septembrie 2011)

[20] Pagina oficială Layar: <<<http://www.layar.com/>>>, accesată ultima oară la 29 decembrie

2011

[21] Pagina oficială LibreGeoSocial: <<<http://www.libregeosocial.org/>>>, accesată ultima oară la 29 decembrie 2011

[22] Imagine ilustrativă GPS Tracking:

<<https://lh3.ggpht.com/76rrxYp99kcUn8h2a82m7z5h12NpY9ACLstdbdE9vzHw1fyxyccknpVH_yhQYn_p3ncg>>, accesată ultima oară la 3 ianuarie 2012

[23] Imagine ilustrativă Tracking:

<<https://lh6.ggpht.com/2qeFRcjxtf4h4LIRDLR9NQonqWb535FbvbcEBd_PdtxhoY6u9Cm8P8Ac1JdauaZM3mM>>, accesată ultima oară la 3 ianuarie 2012

[24] Imagine ilustrativă Match2blue:

<<https://market.android.com/details?id=de.match2blue&feature=search_result#?t=W251bGwsMSwxLDEsImRILm1hdGNoMmJsdWUiXQ..>>, accesată ultima oară la 3 ianuarie 2012

[25] Imagine ilustrativă Wikitude: <<<http://blackberrysync.com/wp-content/uploads/2012/04/Capture-on-04-30-2012-08-55-15-400x300.jpg>>>, accesată ultima oară la 30 iunie 2012

[26] Imagine ilustrativă LibreGeoSocial:

<<https://market.android.com/details?id=com.LibreGeoSocial.App&feature=also_installed>>
accesată ultima oară la 3 ianuarie 2012

[27] Android Market, Pagina Layar,

<<https://market.android.com/details?id=com.layar&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5sYXlhcjId>>, accesată ultima oară la 3 ianuarie 2012

[28] Imagine ilustrativă Junaio: <<<https://market.android.com/details?id=com.metaio.junaio>>>
accesată ultima oară la 3 ianuarie 2012

[29] "Scan the World" with the new junaio 3.0 Augmented Reality Browser, (în format electronic la: <<<http://www.metaio.com/press/press-release/2011/scan-the-world-with-the-new-junaio-30-release/>>>, accesat ultima oară la 3 ianuarie 2012).

[30] Sateliții GPS: <<<http://androidheadlines.com/2010/08/home-gps-fix-for-samsung->

vibrantcapture-phones.html>>, accesat ultima oară la 29 mai 2012.

[31] Miller G., Stroud A., 2012, Professional Android Sensor Programming, Wrox

[32] KML Tutorial (în format electronic la adresa << https://developers.google.com/kml/documentation/kml_tut>>, accesat ultima oară la 30 mai 2012).

[33] KML Reference (în format electronic la adresa << <https://developers.google.com/kml/documentation/kmlreference> >>, accesat ultima oară la 30 mai 2012)

[34] Reto Meier, 2010, Professional Android™ 2 Application Development, Wiley Publishing, Inc., Capitolele 3, 15

[35] Mark Murphy, CWAC'd Up: Alternative Adapters, Android Guys, în format electronic la adresa <<<http://www.androidguys.com/2009/07/29/cwacd-up-alternative-adapters/>

Glosar

ARMD	Advanced RISC Machine11
CPU	Central Processing Unit
RAM	Random access memory
MCP	Multi-Chip Package
GPS	Global Positioning System
SD	Secure Digital
USB	Universal Serial Bus
GPLv3	GNU GENERAL PUBLIC LICENSE, Verision 3
DPI	Dots Per Inch
MCC	Mobile Country Codes
MNC	Mobile Netwok Code
MAC	Media Access Control
SSID	Service Set Identifier
KML	Keyhole Markup Language

PDU

Protocol Data Unit