

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA  
CAMPUS DI CESENA  
SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

# **Applicazione car pooling, condivisione dei posti auto attraverso i social network**

Relazione finale in  
Mobile Web Design

Relatore  
Dott. Mirko Ravaioli

Presentata da  
Mirko Squadrani

Sessione III

Anno Accademico 2012 – 2013



# Indice

<b>PREFAZIONE</b> .....	<b>6</b>
<b>1. INTRODUZIONE</b> .....	<b>8</b>
1.1 IL “CAR POOLING” .....	8
1.2 I SOCIAL NETWORK.....	9
1.3 ANDROID.....	10
<b>2. FASE DI PROGETTAZIONE</b> .....	<b>12</b>
2.1 STRUTTURA DEL PROGETTO.....	12
2.2 HOME.....	13
2.3 INSERIMENTO VIAGGIO.....	14
2.4 RICERCA DI UN VIAGGIO.....	15
2.5 NOTIFICA PASSAGGIO DISPONIBILE.....	16
2.6 LOGICA DI PROGETTAZIONE.....	17
2.7 FACEBOOK APIs.....	17
2.7.1 Graph API.....	18
2.7.2 FQL (Facebook Query Language) API.....	20
2.7.3 Chat API.....	20
2.7.4 Keyword Insights API.....	21
2.7.5 Public Feed API.....	21
2.7.6 Ads API.....	21
<b>3. FASE DI IMPLEMENTAZIONE</b> .....	<b>22</b>
3.1 ACTIVITY DEL PROGETTO.....	22
3.1 INTERROGAZIONI FQL.....	23
3.2 SERVIZIO DI NOTIFICA.....	25
<b>4. CONCLUSIONI</b> .....	<b>29</b>
<b>BIBLIOGRAFIA</b> .....	<b>30</b>
<b>RINGRAZIAMENTI</b> .....	<b>31</b>



## **PREFAZIONE**

L'obiettivo di questa tesi consiste nello sviluppare un'applicazione su piattaforma android che permetta di organizzare il car pooling attraverso i social network.

Attualmente esistono altri software, però sono basati su un sistema privato di gestione degli utenti e viaggi; mentre l'idea che mi è stata data dal mio relatore è quella di usare qualcosa che esiste già come un social network, tipo Facebook, Google+, Twitter, ecc... e permettere tramite un semplice aggiornamento di stato creato dall'utente, di informare gli stessi e poter organizzare un viaggio assieme ad altre persone che usano l'applicazione.

Nel mio caso ho usato Facebook come social network perché è il più diffuso nel mondo ed è conosciuto da tutti.

Per facilitare l'integrazione della piattaforma con le applicazioni Android, ho usato il Facebook SDK, il quale mette a disposizione gli strumenti per effettuare il login con l'autenticazione di Facebook, leggere e scrivere con le *Facebook APIs* ed il supporto per gli elementi dell'interfaccia utente come raccoglitori e finestre di dialogo.

Per quanto riguarda Android attualmente l'ultima versione sviluppata è la 4.4 denominata "KitKat". Questo sistema operativo si può trovare su apparecchiature di vario genere, ma soprattutto non è legato a nessuna casa di produzione di dispositivi mobili. Ogni produttore inoltre personalizza la propria versione del sistema operativo, specialmente dal punto di vista grafico. Questo è possibile in quanto è un sistema open-source, quindi libero dal copyright.



# **1. INTRODUZIONE**

Prima di procedere con la fase di progettazione del software bisogna sapere quali sono gli argomenti trattati dalla tesi: da un'introduzione al car pooling, alla definizione dei social network, fino ad un discorso chiarificatore sulla piattaforma android.

## **1.1 IL “CAR POOLING”**

Il termine inglese “car pooling”, corrisponde al termine italiano “auto di gruppo”, indica una modalità di trasporto che consiste nella condivisione di automobili private tra un gruppo di persone, con il fine principale di ridurre i costi del trasporto, inquinare meno ed aumentare la socializzazione.

Nel car pooling uno o più soggetti coinvolti mettono a disposizione il proprio veicolo, eventualmente alternandosi nell'utilizzo, mentre gli altri contribuiscono con un adeguato contributo in denaro a coprire una parte delle spese sostenute dai proprietari delle automobili. Tale modalità di trasporto è diffusa in ambienti lavorativi o universitari, dove diversi soggetti, che percorrono la medesima tratta nella stessa fascia oraria, spontaneamente si mettono d'accordo per viaggiare assieme.

Questa pratica del condividere l'auto è maggiormente diffusa nei paesi del nord Europa e negli Stati Uniti d'America dove esistono associazioni specifiche e dove la attività è prevista anche nella segnaletica stradale, mentre trova tuttora moderata applicazione in Italia.

Nel 2009 anche Autostrade per l'Italia diede luogo ad una iniziativa per incentivare al car pooling, effettuando delle riduzioni sull'importo del pedaggio autostradale.

Questa pratica ha visto negli ultimi anni un incremento notevole, grazie alla diffusione di numerose piattaforme web (carpooling.it, mitfahrgelegenheit.de, roadsharing.com) che consentono a chi cerca ed a chi offre un passaggio di incontrarsi e definire al meglio i dettagli organizzativi del viaggio. Le piattaforme web consentono, in particolare, di superare uno dei principali ostacoli della diffusione della pratica del car pooling, dato dalla naturale diffidenza nel condividere un viaggio con degli sconosciuti. Però grazie ad un sistema di commenti e valutazioni presente su alcune piattaforme, è possibile infatti conoscere in anticipo il comportamento abituale del proprio eventuale compagno di viaggio, facilitando il superamento della diffidenza e l'accettazione della condivisione dello spostamento.

Di recente si sono diffuse anche molte applicazioni per smartphones, iphone e tablets.

## 1.2 I SOCIAL NETWORK

Al giorno d'oggi, quando si vogliono far entrare in contatto persone sconosciute si usano i social network, ma cosa sono?

Un social network è una rete sociale di un qualsiasi gruppo di persone connessi da diversi legami sociali. Per gli esseri umani i legami vanno dalla conoscenza casuale, ai rapporti di lavoro, ai vincoli familiari. Le reti sociali sono spesso usate come base di studi interculturali in sociologia, in antropologia ed in etologia.

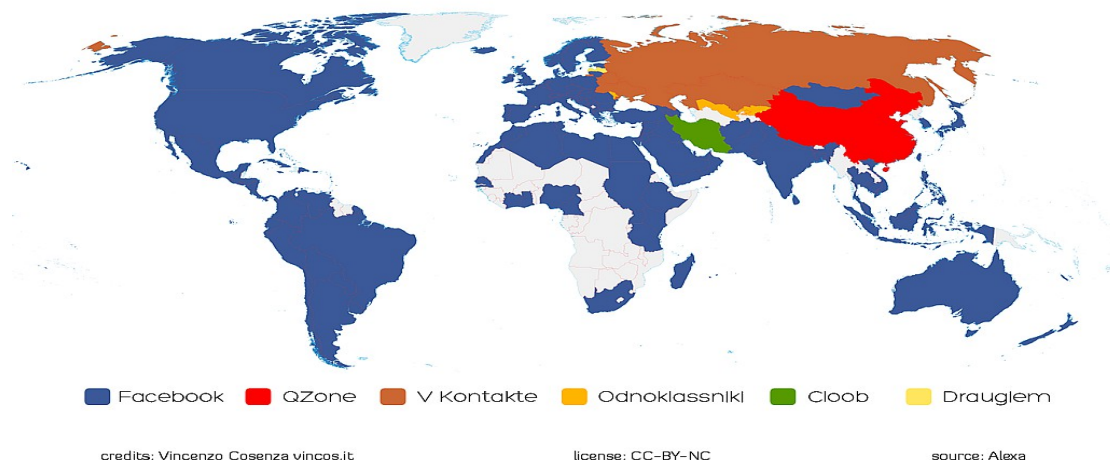
Una rete sociale, esprimibile sempre mediante un grafo, è caratterizzata da una sua propria densità. Se con i nodi di un determinato grafo si identificano gli individui e con gli archi i legami che tra essi si instaurano, allora la densità di una rete può rendere un'idea di quanto sia efficiente l'interscambio relazionale tra i vari elementi della rete stessa.

L'antropologo britannico Robin Dunbar teorizzò che il “*numero di Dunbar*”, conosciuto anche come regola dei 150, afferma che le dimensioni di una rete sociale in grado di sostenere relazioni stabili sono limitate, appunto, a circa 150 membri.

Un servizio di social network consiste in una struttura informatica che gestisce nel Web le reti basate su relazioni sociali ed è anche un tentativo di violare la “*regola dei 150*”. La rete delle relazioni sociali che ciascuno di noi tesse ogni giorno, in maniera più o meno casuale, nei vari ambiti della nostra vita, si può così “materializzare”, organizzare in una “mappa” consultabile, e arricchire di nuovi contatti.

Invece si definiscono siti di reti sociali (*social network sites*) quei servizi web che permettono: la creazione di un profilo pubblico o semi-pubblico, l’articolazione di una lista di contatti, la possibilità di scorrere la lista di amici dei propri contatti. Attraverso ciò, questi servizi permettono di gestire e rinsaldare online amicizie preesistenti o di estendere la propria rete di amici.

### WORLD MAP OF SOCIAL NETWORKS December 2013



**Illustrazione 1: Mappa mondiale dei social network, aggiornata a dicembre 2013.**



Il primo social network fu SixDegrees che nacque nel 1997, ma in generale ci fu un'esplosione nel 2003, anno in cui nacquero LinkedIn, Orkut, MSN, Myspace e Facebook.

### 1.3 ANDROID

E' un software stack, cioè un insieme di programmi *open-source* che lavorando assieme producono un risultato, per un'ampia gamma di dispositivi mobili, ed è anche un progetto open-source guidato da Google Inc.

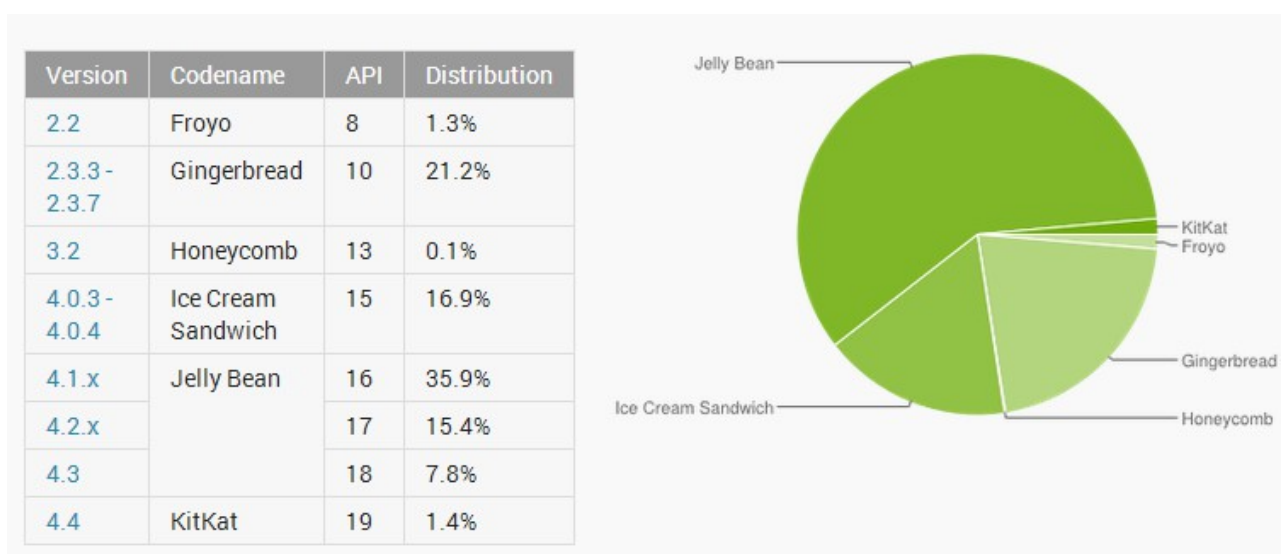
Esso ha come obiettivo quello di fornire tutto ciò di cui un operatore, un vendor di dispositivi o uno sviluppatore necessita, per raggiungere i propri scopi.

Android Inc. è stata fondata nell'ottobre del 2003 da Andy Rubin, Rich Miner, Nick Sears e Chris White. Successivamente nel 2005 Google Inc. acquisì l'azienda in vista del fatto che la società di Mountain View desiderava entrare nel mercato della telefonia mobile. E' in questo periodo che il team di Rubin cominciò a sviluppare un sistema operativo per dispositivi mobili, basato su kernel linux. La presentazione ufficiale del "robotto verde" avvenne nel 2007, quando le principali aziende nel mondo della telefonia diedero origine alla Open Handset Alliance (OHA).

Oltre a Google, troviamo infatti produttori di dispositivi come Motorola, Sprint-Nextel, Samsung, Sony-Ericsson e Toshiba; ed operatori mobile come Vodafone, T-Mobile e costruttori di componenti come Intel e Texas Instruments.

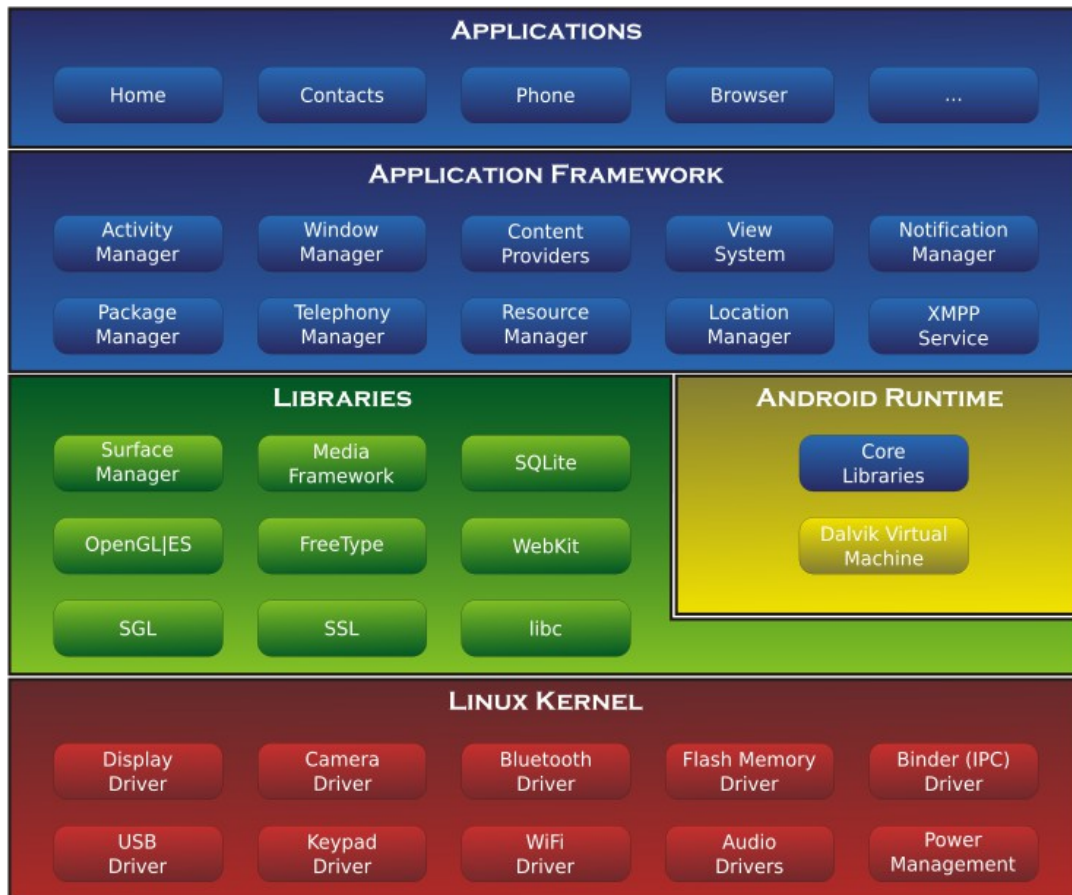
Sempre nel 2007 esce la prima versione del Software Development Kit (SDK), il quale consentì agli sviluppatori di iniziare a toccare con mano la nuova piattaforma e realizzare le prime applicazioni sperimentali, che dal 2008 sono state testate sul primo dispositivo reale, ovvero il G1 della T-Mobile.

Qui sotto è illustrata la diffusione delle varie versioni del sistema Android.



**Illustrazione 2: Diffusione sistema Android, durante il periodo di una settimana del 8 Gennaio 2014**

Il sistema operativo Android è basato sul kernel Linux 2.6 e 3.x, con middleware (cioè insieme di programmi che fungono da intermediari tra le applicazioni), librerie e API (Application Programming Interface). L'Android Developer Tools (ADT) supporta principalmente il linguaggio Java, ma permette di usare anche il C e il C++.



**Illustrazione 3: Schema dell'architettura del sistema**

Per eseguire questi programmi, invece di usare la JVM (Java Virtual Machine), per ottimizzare al massimo l'uso delle risorse nei dispositivi, è stata adottata una propria virtual machine che prende il nome di Dalvik (nome della città islandese del suo inventore), che esegue i Dalvik dex-code (Dalvik Executable), ottenuti a loro volta, in fase di compilazione (building), a partire da file “.class” di bytecode java. Il meccanismo di generazione del codice nella DVM viene detto *register based* (orientato ai registri) a differenza di quello della JVM di tipo *stack based* (orientato all'utilizzo dello stack). Grazie a ciò si ottiene una riduzione di circa il 30% delle operazioni. Per esempio, considerando l'operazione  $c=a+b$ , con la jvm si dovrebbero eseguire quattro comandi, invece con la dvm basta eseguire un comando: “*add a,b,c*”.

## 2. FASE DI PROGETTAZIONE

In questo capitolo viene eseguita un'analisi del progetto fino ai minimi dettagli, poi in seguito verrà spiegato come collegare un applicazione a Facebook.

Per far interagire questo progetto al social network, è necessario usare il pacchetto Facebook SDK, il quale è la via più semplice per integrare un applicazione per Android con la piattaforma social. Esso mette a disposizione molti strumenti, per effettuare il login con l'autenticazione di Facebook, leggere e scrivere tramite le APIs ed anche un supporto per gli elementi dell'interfaccia utente.

### 2.1 STRUTTURA DEL PROGETTO

L'idea del progetto è quella di realizzare un applicazione per android, che permetta agli utenti dei social network di effettuare il “car pooling” con altri utilizzatori di questo software.

L'applicazione è strutturata principalmente in quattro parti:

- Home
- Inserimento di un viaggio
- Ricerca di un passaggio
- Servizio di notifica di un passaggio disponibile

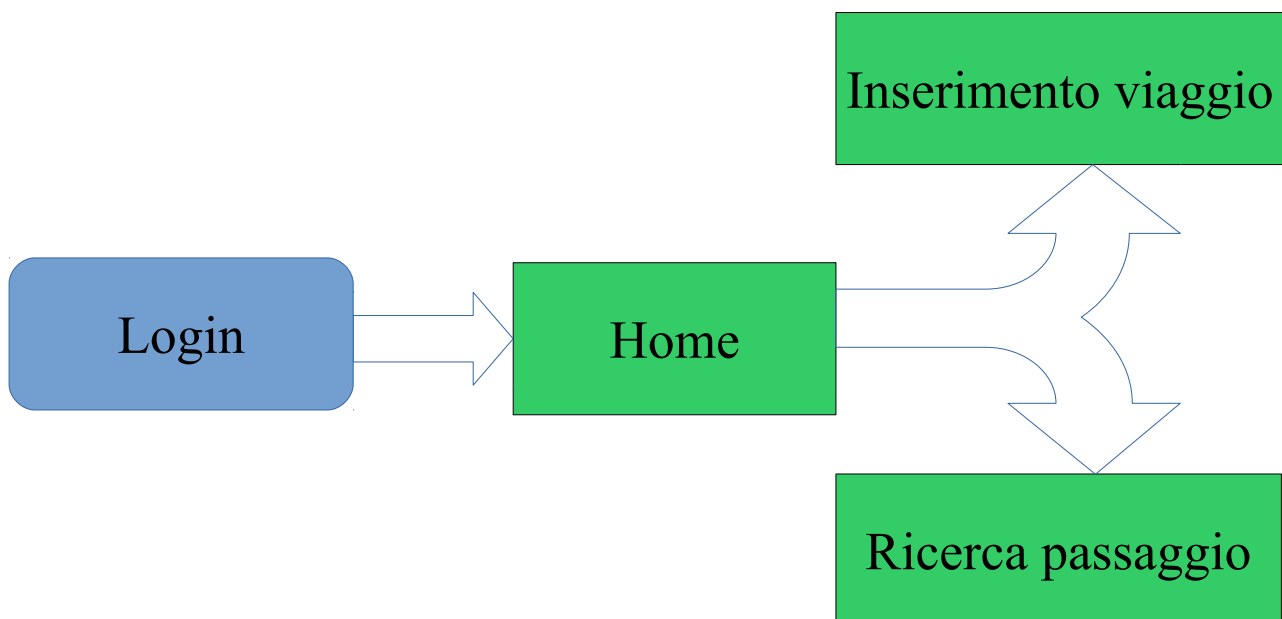


Illustrazione 1: Schema dell'applicazione

## 2.2 HOME



Illustrazione 2: Home prima del login

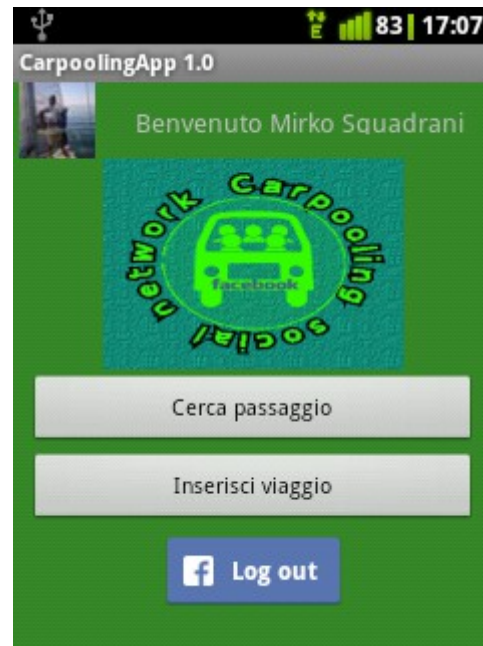


Illustrazione 3: Home dopo il login

Questa è l'*activity* principale dell'applicazione, ed è strutturata in questo modo.

La figura a sinistra, rappresenta la prima schermata che vediamo quando lanciamo l'applicazione se non è ancora stato effettuato l'accesso. Per usarla è, quindi, necessario avere una connessione a internet ed essere registrati a Facebook. Sulla destra invece è rappresentata la home dopo il login, e sono quindi visibili i pulsanti delle varie funzionalità. Per disconnettersi, invece, basterà premere il pulsante blu di “Log out”.

La Facebook Login è un metodo personalizzato e affidabile per l'autenticazione degli utenti. In quanto usa il sistema di sicurezza “OAuth2.0”, un protocollo *open-source*, che effettua l'autenticazione dell'utente.

Appena viene effettuato il login, si accede al profilo pubblico, alla lista degli amici ed a brevi informazioni generiche. Ogni altra informazione è negata per la tutela della privacy. Quindi per accedere ad ulteriori informazioni personali deve essere richiesto uno specifico permesso all'utente. In questo caso è stata richiesta l'autorizzazione per poter accedere: agli aggiornamenti di stato degli amici, di poter leggere e pubblicare post ed infine di accedere all'email personale.

Una classe fondamentale per effettuare il login in modo veloce è usare la *LoginButton*. Essa serve ad aprire o chiudere la sessione del utente, visualizzando allo stesso tempo il testo corretto nel pulsante di accesso.

Per gestire la grande complessità dei cambiamenti di stato della sessione, si usano le due classi: *UiLifecycleHelper* e *Session.statusCallback*. La prima viene usata da

metodi pubblici del ciclo di vita dell'activity per creare, salvare, cancellare e ripristinare la sessione corrente di Facebook. La seconda, invece, rimane in ascolto monitorando ogni cambiamento di stato. Può anche sovrascrivere il metodo “*call()*” per aggiornare l'interfaccia grafica.

Premendo il pulsante “Inserisci viaggio” si aprirà una seconda *activity*, il cui funzionamento verrà spiegato nel prossimo paragrafo.

## 2.3 INSERIMENTO VIAGGIO

Come già accennato in precedenza, premendo il pulsante “Inserisci viaggio” si aprirà una *activity* la cui funzionalità è quella di poter programmare un itinerario da una città ad un'altra.

The image shows a mobile application interface for entering travel details. The title bar at the top reads "Inserimento viaggio". Below the title bar, there are several input fields: "Tipo passaggio" with a dropdown menu showing "offerta"; "Città di partenza" with a text input field containing "Rimini"; "Città di arrivo" with a text input field containing "Cesena"; "Data e Ora di partenza" with two separate input fields for the date "20/1/2014" and the time "10:21"; and "Numero di posti liberi" with a dropdown menu showing "3". The background of the form is green.

**Illustrazione 4: Inserimento viaggio**

L'activity accetta in input questi parametri: la città di partenza e di arrivo, la data e l'ora di partenza ed il numero di posti disponibili. Nella figura qui sopra viene mostrato un esempio di inserimento di un viaggio.

Esso può essere pubblicato in un aggiornamento di stato di Facebook, con una semplice sintassi in formato JSON, premendo la voce di menu “Pubblica”. In questo modo l'applicazione può rileggere i dati, rielaborarli e mostrarli all'utente in modo corretto. La figura sottostante, rappresenta un aggiornamento di stato, generato dal sistema.

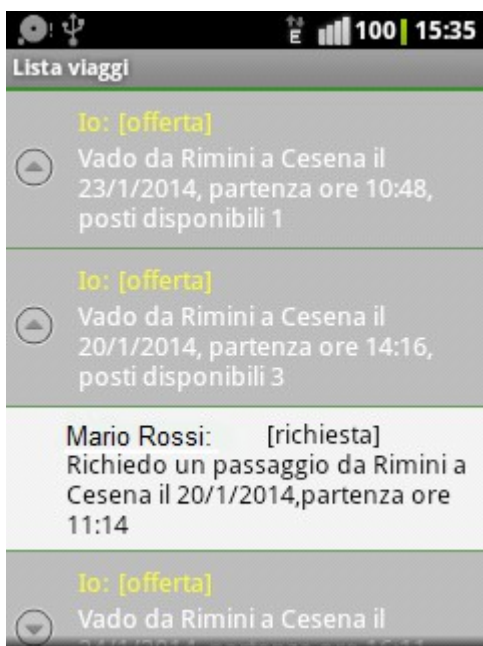


**Illustrazione 5: Aggiornamento di stato pubblicato dall'applicazione**

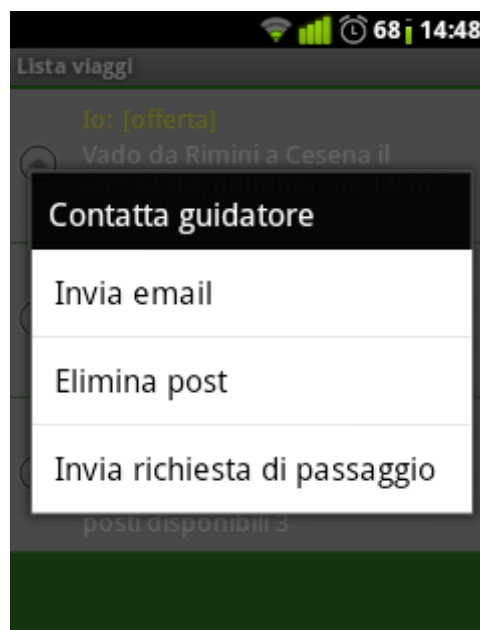
In fine premendo il pulsante “Cerca passaggio” viene aperta una seconda activity il cui meccanismo verrà spiegato nel paragrafo seguente.

## 2.4 RICERCA DI UN VIAGGIO

Come già accennato in precedenza, premendo il pulsante “Cerca passaggio” si aprirà questa activity, il quale permette di cercare un utente che fa car pooling.



**Illustrazione 6: Lista viaggi trovati**



**Illustrazione 7: Finestra di dialogo per contattare il guidatore**

Nella figura a sinistra è mostrato il risultato dello smistamento degli aggiornamenti di stato, restituiti dalla interrogazione. Durante l'analisi dei viaggi, il sistema è in grado di riconoscere diverse combinazioni:

- Quando un utente fa il mio stesso tragitto.



- Quando la destinazione del mio tragitto coincide con la destinazione di un utente.
- Quando esistono due utenti che hanno un tragitto che è tra la mia città di partenza e quella di arrivo; per esempio se il mio tragitto è Faenza – Cesena, ed un utente va da Faenza a Forlì, mentre un altro va da Forlì a Cesena, li posso inserire sotto il mio viaggio.

Un'altra funzionalità permette, tenendo premuto un elemento della lista, di far apparire una finestra di dialogo per contattare il guidatore (come nella figura a destra), in cui si possono effettuare diverse scelte:

- Inviare un e-mail: ovvero creare un messaggio che di default verrà inviato su Facebook, ma il destinatario può comunque essere cambiato.
- Cancellare il post selezionato: permette di eliminare i propri post, ma ovviamente, non quelli degli altri.
- Inserire una richiesta di passaggio: apre l'activity “inserimento viaggio” già precompilata con i parametri del itinerario selezionato, il quale permette di pubblicare un post per la richiesta di passaggio.

## 2.5 NOTIFICA PASSAGGIO DISPONIBILE

Questo servizio permette di ricevere una notifica di sistema, quando viene trovato un utente che fa la mia stessa strada; essa è generata da un API di Android chiamata *NotificationManager*. Cliccando sulla notifica è possibile inviare una richiesta di passaggio, che verrà poi inserita in lista sotto l'offerta corrispondente.



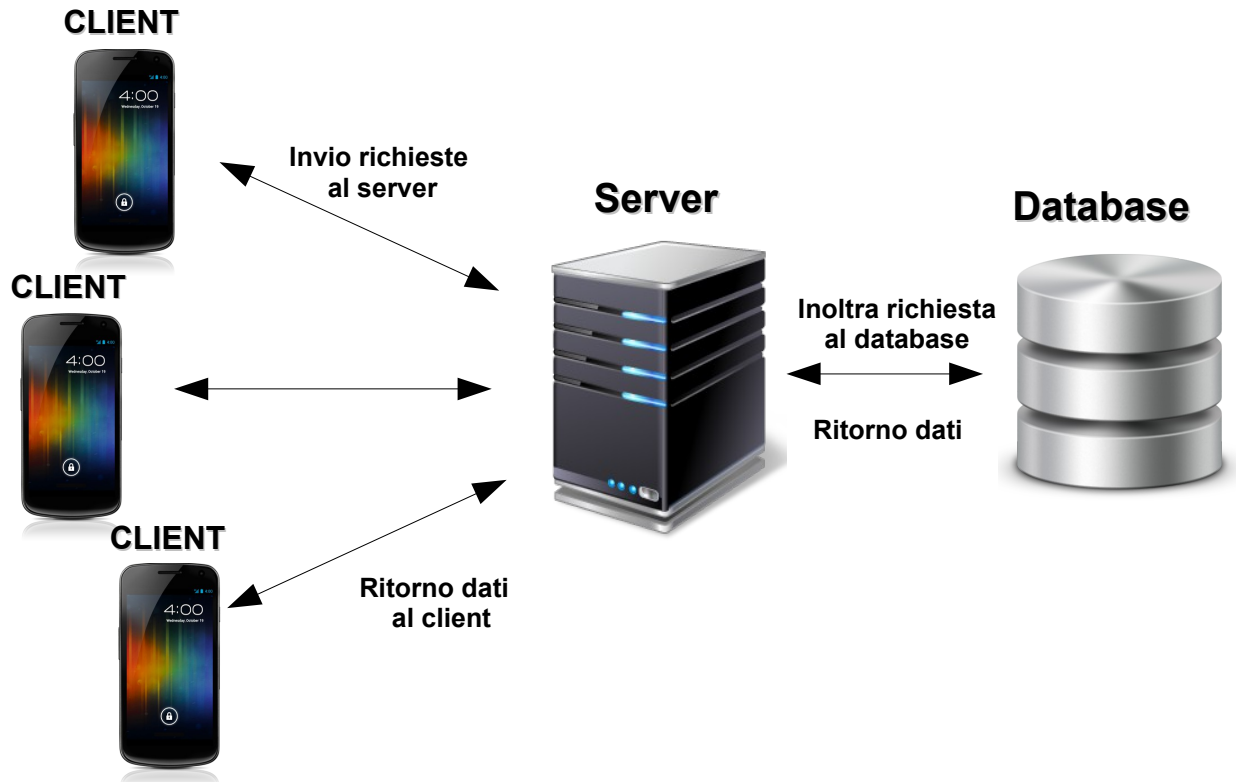
**Illustrazione 8: Esempio notifica di sistema dell'applicazione**

## 2.6 LOGICA DI PROGETTAZIONE

L'applicazione è caratterizzata da due parti principali: lato client e lato server.

Il lato server si occupa di gestire le interrogazioni che vengono effettuate dall'applicazione al server; questa parte viene interamente gestita da Facebook.

Il lato client ha il compito di ricevere i dati del server, gestirli in modo corretto per l'esatto funzionamento all'interno dell'applicazione ed esporli all'utente.



**Illustrazione 9: Schema client – server**

In figura è illustrato il procedimento di base per effettuare una richiesta al server, che a sua volta apre una connessione al database ed esegue una specifica interrogazione, in seguito, vengono ricevuti i dati dal database e restituiti al client.

## 2.7 FACEBOOK APIs

Le API (Application Programming Interface) sono una raccolta di librerie che formano un set di strumenti specifici per lo svolgimento di un determinato compito all'interno di un certo programma. In questo caso aiutano ad interfacciarsi con la piattaforma di Facebook. Esistono diversi tipi di API, che ora andremo ad elencare:

- Graph API
- FQL API
- Chat API

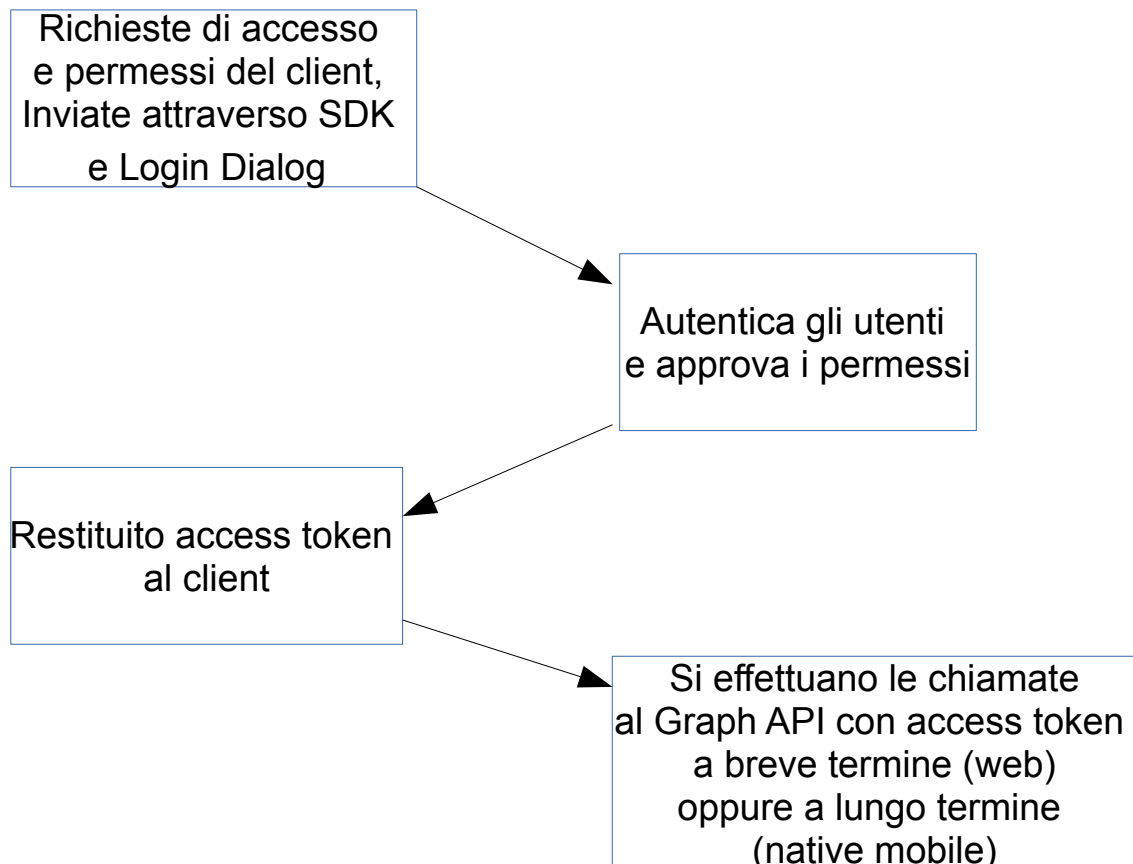


- Keyword Insights API
- Public Feed API
- Ads API

### 2.7.1 Graph API

La Graph API sono il principale modo per le applicazioni di leggere e scrivere nel grafo sociale di Facebook. Questa api a sua volta usa un insieme di librerie di basso livello, raggruppate in un pacchetto chiamato HTTP-Based API, il quale rende possibile usare i dati delle query, inserire nuovi post, caricare foto e tante altre cose.

Molte richieste effettuate dalla Graph API, richiedono l'uso di un *access token*, che è una stringa automaticamente generata dall'applicazione quando si effettua il login; esso serve per garantire un accesso temporaneamente sicuro alle Facebook APIs, e ad identificare l'applicazione dell'utente. Di seguito si presenta uno schema per spiegare l'uso del *access token* con le APIs.



**Illustrazione 10: Schema delle chiamate alle API con access token, da parte dei client nativi o web.**

Le APIs sono composte da: **nod**i (come un utente, una foto, una pagina, un commento, ecc...), **archi** (come le foto di una pagina, oppure i commenti di una foto,

ecc...), e **campi** (come la data di compleanno dell'utente, oppure il nome di una pagina, ecc...).

Tutti i nodi e gli archi nella Graph API possono essere letti semplicemente con una richiesta HTTP GET. In questo caso viene richiesto il profilo pubblico dell'utente (nome, cognome e foto), e quindi eseguita questa direttiva:

```
GET graph.facebook.com/{user-id}
```

dove il parametro *{user-id}* si deve sostituire con l'id di un utente qualsiasi oppure con il nodo speciale *"/me"*, che permette di ottenere il mio id personale.

La risposta che si riceve dal sistema, è una stringa JSON (Javascript Object Notation) che è un semplice formato, molto utilizzato per lo scambio di dati fra applicazioni client – server, che rispetta la forma *{ "chiave" : "valore" }*:

```
{
  "id": "1029121181",
  "name": "Mirko Squadrani",
  "first_name": "Mirko",
  "last_name": "Squadrani",
  "link": "https://www.facebook.com/mirko.squadrani",
  "gender": "male",
  "timezone": 1,
  "locale": "it_IT",
  "verified": true,
  "updated_time": "2013-12-19T15:15:24+0000",
  "username": "mirko.squadrani"
}
```

Molti nodi nella Graph API possono avere archi che potrebbero essere pubblicati (come una foto o un post). Per inviarli basta eseguire una richiesta HTTP POST con i relativi parametri. Il risultato sarà un valore booleano: *true* se l'esito è positivo oppure *false* altrimenti. Qui sotto è mostrato un comando per inviare un aggiornamento di stato:

```
POST graph.facebook.com
  /me/feed?
  message={message}&
  access_token={access-token}
```

I parametri *{message}* e *{access-token}* indicano rispettivamente il contenuto del post e la stringa *access token* generata dall'applicazione.

E' possibile anche eliminare i nodi sempre usando la Graph API effettuando la richiesta HTTP DELETE, in questo modo:

```
DELETE graph.facebook.com
  /{node-id}?
  access_token={access-token}
```

Il parametro *{node-id}* rappresenta l'id del nodo (post, foto,...) che si vuole eliminare.

### **2.7.2 FQL (Facebook Query Language) API**

Facebook Query Language, abilita all'uso dell'interfaccia *SQL-style* per eseguire interrogazioni su i dati mostrati dalla Graph API. Questo pacchetto offre alcune funzionalità avanzate, come la possibilità di riutilizzare i risultati di una interrogazione in un'altra (cioè subquery, query innestate o nidificate).

Per implementare i servizi che l'applicazione dispone, bisogna eseguire alcune interrogazioni al database di Facebook; più dettagliatamente si possono distinguere quattro tipi di query:

- Interrogazione sui post personali: serve per ottenere i post pubblicati nel diario (stream) del utente.
- Interrogazione sui post degli utenti: si ottengono i post pubblicati da i soggetti che usano l'applicazione.
- Interrogazione sul profilo personale: viene eseguita appena si effettua il login, tramite il metodo “*loginButton.setUserInfoChangedCallback()*”, il quale esegue in modo implicito una interrogazione sul profilo pubblico usando la Graph API.
- Interrogazione sul profilo degli utenti: si ottiene il profilo pubblico dei soggetti che usano l'applicazione.

### **2.7.3 Chat API**

Tramite questa libreria è possibile integrare la chat di Facebook con i prodotti Web-based, desktop, o smartphone. La connessione al client di messaggistica istantanea (IM) avviene tramite un servizio Jabber/XMPP.

Extensible Messaging and Presence Protocol (XMPP), noto precedentemente come “Jabber”, è un insieme di protocolli per la messaggistica istantanea e presenza basato su XML. Jabber è anche un servizio IM che implementa i protocolli XMPP.

### **2.7.4 Keyword Insights API**

Questa api permette di analizzare tutti i post di Facebook, quindi si è abilitati ad eseguire query globali e ricerche anonime su certi termini che gli utenti scrivono nei post. Però l'accesso a questa api è limitato ad una ristretta cerchia di editori e per essere usata serve l'approvazione di Facebook.

### ***2.7.5 Public Feed API***

Tramite questa libreria si può accedere alla bacheca pubblica degli aggiornamenti di stato di una pagina o di un utente, per postarne di nuovi o fare ricerche a scopo statistico. Però anche questo canale è ristretto ad un numero limitato di editori, e per usarlo bisogna sempre ottenere l'approvazione di Facebook.

### ***2.7.6 Ads API***

Questa api permette di creare ed accedere ai propri annunci pubblicitari programmati organizzati in campagne. Ognuna di queste campagne contiene dei gruppi di annunci, che possono contenere a loro volta delle offerte, obiettivi e annunci creativi. Quest'ultimi definiscono l'immagine dello stesso.

### **3. FASE DI IMPLEMENTAZIONE**

La fase di lavoro che si introduce riguarda la vera e propria realizzazione dell'applicazione. In questo progetto il linguaggio di programmazione usato è il Java, in quanto si utilizzano le librerie di Android SDK.

Il codice scritto verrà riportato solo se ritenuto strettamente necessario, verrà piuttosto riportata una descrizione in linguaggio naturale che è più comprensibile da tutti.

#### **3.1 ACTIVITY DEL PROGETTO**

In fase di progettazione si è parlato tanto di activity, ma cosa sono?

Un activity è un componente dell'applicazione che serve per interagire con l'utente. Ognuna di essa è definita da una finestra nella quale è disegnata l'interfaccia utente. Un'applicazione consiste di solito in molte activity che si alternano.

In questo caso c'è un activity principale chiamata “MainActivity”, che viene lanciata appena si apre l'applicazione. Mentre le altre vengono eseguite a seguito di un'azione (come la pressione di un pulsante).

Ogni volta che una seconda activity parte, viene messa nello stack (il “back stack” è una pila di dati “last in, first out”), mentre quella precedente viene fermata e mantenuta nello stack. Quando l'utente preme il tasto “Back”, l'activity corrente viene rimossa dallo stack (e distrutta) ed in seguito ripristinata quella precedente.

Ogni volta che un activity cambia stato, l'evento viene notificato attraverso i metodi del ciclo di vita della activity (per esempio, onCreate, onStart, onResume, ecc...).

Per spiegarmi meglio, qui di seguito è mostrato uno schema del loro ciclo di vita.



- Interrogazione per i post degli utenti:

```
SELECT post_id, source_id, message
FROM stream
WHERE source_id IN
(SELECT uid FROM user WHERE is_app_user = 1 AND uid IN
    (SELECT uid2 FROM friend WHERE uid1 = me()))
AND app_id = 549370771821285
AND strpos(lower(message), '{') >= 0
ORDER BY created_time DESC LIMIT 50
```

- Interrogazione sul profilo pubblico personale: questa è una query, eseguita in modo implicito dal metodo *setUserInfoChangedCallback()*.

```
loginButton.setUserInfoChangedCallback(
new LoginButton.UserInfoChangedCallback() {
    @Override
    public void onUserInfoFetched(GraphUser user) {
        MainActivity.this.user = user; //Dati profilo utente
        updateUI(); //Aggiorna interfaccia utente
    }
});
```

- Interrogazione sul profilo pubblico degli utenti: con questa query vengono ricavati i dati degli utenti che usano l'applicazione.

```
SELECT username, email, name, first_name, last_name, uid, pic_small
FROM user
WHERE uid IN
(SELECT uid FROM user WHERE is_app_user = 1 AND uid IN
    (SELECT uid2 FROM friend WHERE uid1 = me()))
```

In fine si deve precisare che i risultati delle interrogazioni, effettuate sulla tabella “Stream”, sono stati limitati a 50 per ottimizzare i tempi di elaborazione, visto il gran numero di post che le persone pubblicano ogni giorno.

### 3.2 SERVIZIO DI NOTIFICA

Innanzitutto un servizio in android è un componente dell'applicazione che può eseguire solo operazioni in background e quindi non possiede un'interfaccia grafica.

Un servizio può essere essenzialmente di due tipi:

- Started

È un servizio che viene fatto partire da un'activity richiamando il metodo `startService()`. Appena viene lanciato, rimane in esecuzione in background, per un tempo indefinito, fino a quando non viene fermato. Di solito esegue una sola operazione per volta e non può restituire un risultato al chiamante.

- Bound

È un servizio che viene lanciato con il metodo `bindService()`. Esso offre un'interfaccia client-server in cui ogni componente può: interagire con il servizio, inviare richieste, ottenere risultati, sempre passando attraverso un processo di comunicazione chiamato IPC.

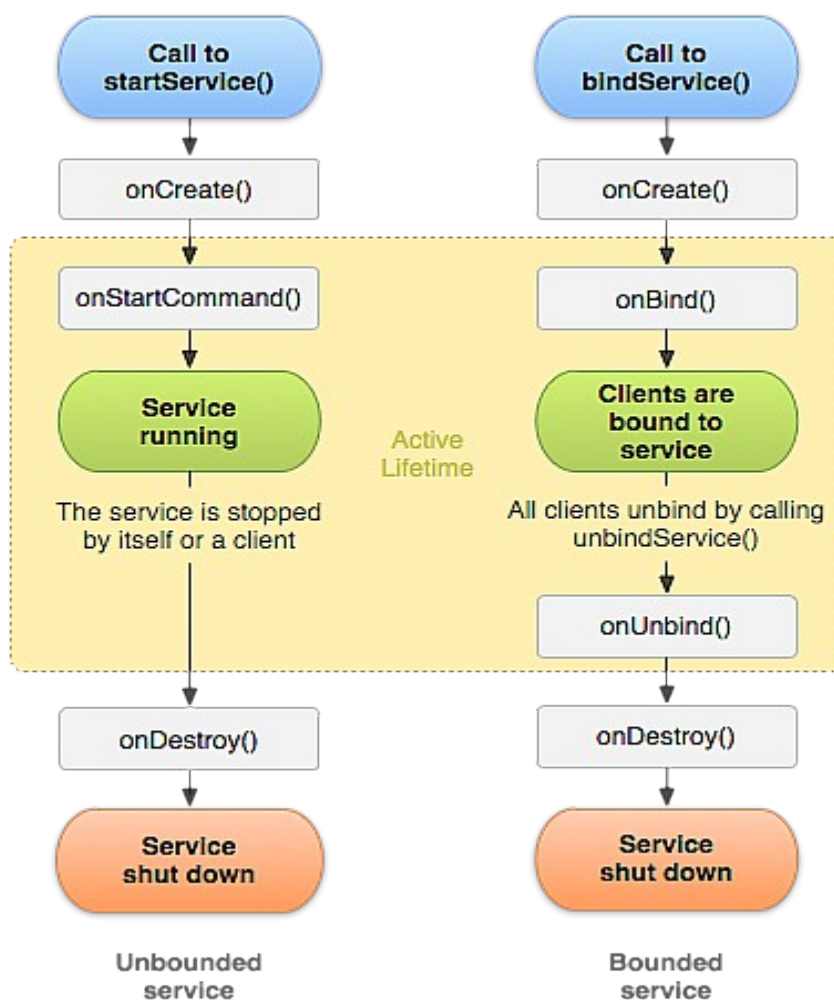


Illustrazione 2: Ciclo di vita dei due tipi di servizio.



Il primo diagramma a sinistra rappresenta quello del primo tipo, cioè “Started”, invece il secondo diagramma a destra è quello di tipo “Bound”. I vari cambiamenti di stato vengono notificati attraverso i metodi pubblici del ciclo di vita del servizio (per esempio, onCreate, onStartCommand, onDestroy, ecc...).

In questo progetto, l'applicazione usa un servizio del primo tipo, che resta sempre attivo, e quando trova un risultato invia immediatamente una notifica di sistema all'utente.



## **4. CONCLUSIONI**

In quest'ultimo capitolo si conclude che il lavoro è stato svolto con qualche difficoltà iniziale, in quanto si creavano sempre problematiche nuove, ma poi sono state tutte superate con l'aiuto di documentazioni online.

Il progetto è stato finito e testato su una versione di android, ed è completamente funzionante, anche se andrebbe ottimizzato. Il tempo totale trascorso per completare l'applicazione è stato di circa due mesi.

L'obiettivo era quello di creare un car pooling attraverso i social network, come Facebook, ed è stato raggiunto.

In futuro si potrebbero effettuare ulteriori miglioramenti, per velocizzare il software durante la fase di smistamento dei post, e collegando la chat di Facebook. Successivamente si dovrebbe migliorare la grafica aggiungendo anche qualche animazione.

Infine si potrebbe sviluppare il progetto su altre piattaforme, come IOS (Apple) e Windows mobile (Microsoft).

## **BIBLIOGRAFIA**

- Wikipedia – car pooling

[http://it.wikipedia.org/wiki/Car\\_pooling](http://it.wikipedia.org/wiki/Car_pooling)

- Facebook developers – Getting Started with the Facebook SDK for Android

<https://developers.facebook.com/docs/android/getting-started/>

- Android, the world's most popular mobile platform

<http://developer.android.com/about/index.html>

- OAuth2.0 – This is an Internet Standards Track document

<http://oauth.net/2/>

- XMPP service

<http://xmpp.org/about-xmpp/history/>

- Mappa mondiale dei social network

<http://vincos.it/world-map-of-social-networks/>

-Stack Overflow is a question and answer site for professional and enthusiast programmers

<http://stackoverflow.com/>

- JSON (JavaScript Object Notation)

<http://www.json.org/>

## **RINGRAZIAMENTI**

Un grazie va ai miei genitori che mi hanno sempre supportato, permettendomi di finire gli studi.

Un altro grazie va a tutti i miei amici che mi hanno sempre dato un aiuto morale.