

Matricola Studente:**Titolo del Progetto: Let's go** **Data:04/09/12****Introduction:**

Let's go è un'applicazione per creare, condividere e cercare eventi sportivi, feste, fiere e quant'altro. È molto simile ad un social network per eventi. Come tutti i social network si ha la possibilità di aggiungere i propri amici. È possibile condividere con loro informazioni riguardanti gli eventi creati da loro stessi o da altre persone.

Il problema principale che vuole risolvere quest'applicazione è quando ci si trova con i propri amici o si vuole organizzare dove andare e non si sa mai cosa fare. L'applicazione mostra tutti gli avvenimenti che sono nella zona con la possibilità di filtrarli in base ai propri interessi o a quello che si sta cercando in quel determinato momento. L'applicazione permette la sincronizzazione con i propri eventi di facebook per sapere sempre dove andare.

Related work:

L'applicazione prende spunto dagli eventi di facebook e da google local. Da Facebook prende l'idea degli amici e degli eventi privati invece da google local prende l'idea di visualizzare, su una mappa, eventi pubblici dove chiunque può apprendere della loro esistenza. La differenza sostanziale risiede dal tipo e dalle autorizzazioni necessarie per la creazione dell'evento.

www.facebook.com

www.maps.google.it

Usage model:

L'idea è quella di rendere l'applicazione il più facile possibile da utilizzare. Sarà presente una schermata iniziale che chiederà di effettuare un login. Se non si possiede già un account sarà possibile crearne uno fornendo poche e semplici informazioni. Queste due schermate saranno visualizzabili finché non si sarà loggati. Una volta loggati sarà presente una schermata con tre tab. Il primo visualizza l'elenco dei degli eventi a cui si è stati invitati e quelli creati, nel secondo tab ci saranno le informazioni dei propri contatti e nel terzo ci sarà una mappa con gli eventi, con la possibilità di filtrarli secondo i propri interessi. Ci saranno inoltre dei pulsanti e dei menu per creare eventi o invitare amici o aggiornare le informazioni.

La creazione degli eventi verrà effettuata scegliendo alcune opzioni, in particolare sarà possibile scegliere la posizione del evento, cercando su una mappa il luogo e selezionandolo.

Cliccando su un evento sarà possibile avere informazioni più dettagliate rispetto a quelle visualizzate sul tab. e da qui sarà possibile centrarlo sulla mappa o ottenere informazioni stradali se si possiede un navigatore installato sul telefono o aggiungere l'evento al proprio google calendar.

Per gli amici invece sarà possibile invitarli, bloccarli e accettare richieste di amicizia. Sarà possibile inviare una richiesta di amicizia o attraverso l'inserimento dell'email dell'amico o utilizzando la propria rubrica del telefono, o utilizzando i propri contatti facebook.



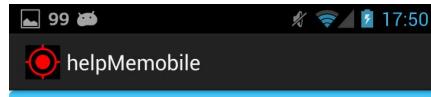
Email

Scrivi la tua mail

Password

Login

Non hai un' account? Registrati



Full Name

Enrico De Guidi

Email

enrico.deguidi@gmail.com

Password

• • • • •

Ripeti Password

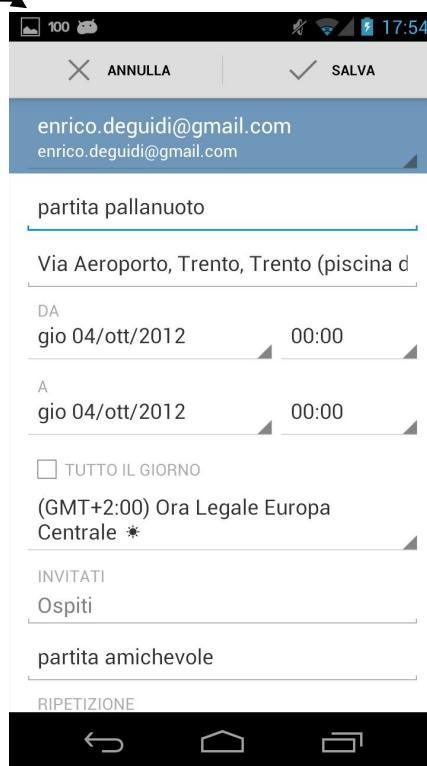
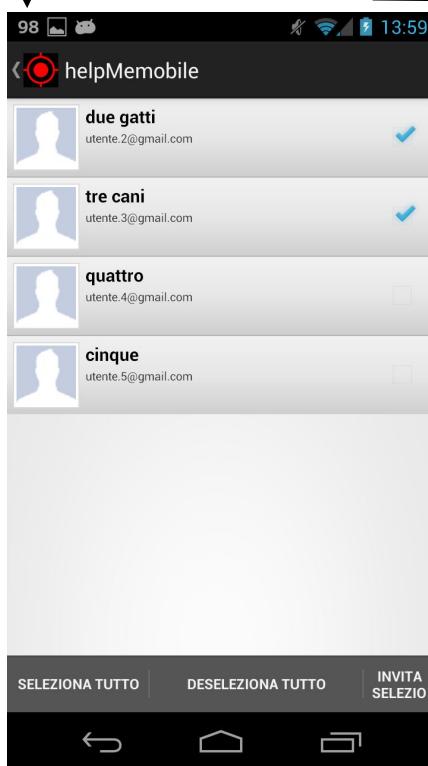
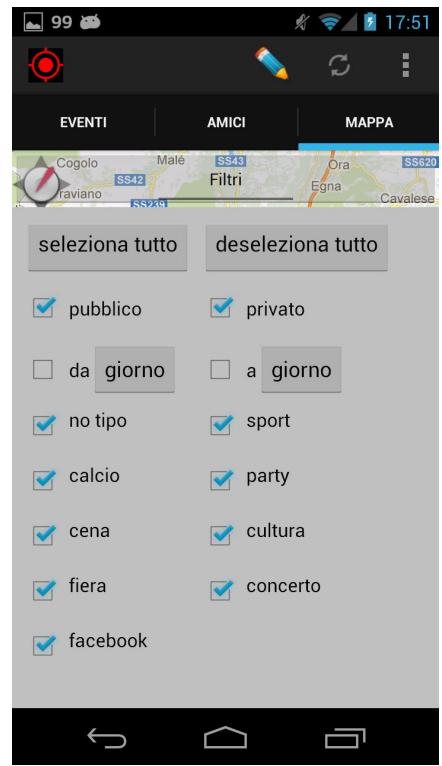
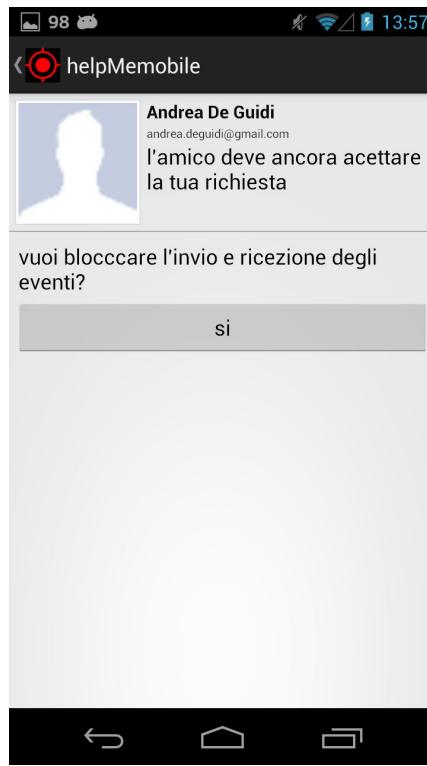
• • • • •

le due password non coincidono

Register

Vai al login!







Architecture design

L'applicazione è composta da due parti fondamentali e una opzionale, tutte e tre posseggono un database che possiede informazioni:

1. Il client è la parte android vera e propria. Il database memorizza le informazioni quali i propri eventi (compresi quelli di facebook se si è sincronizzato l'account), quelli a cui si è stati invitati e quelli pubblici scaricati dalla mappa. Più inoltre le informazioni dell'utente loggato. Una volta effettuato il logout tutte queste informazioni saranno cancellate dal dispositivo.
2. Il server in php implementato. Il database ha memorizzate tutte le informazioni relative agli utenti, alle amicizie e agli eventi ad eccezione di quelli di face book.
3. Il server di facebook. Da dove è possibile ottenere le informazioni relative agli eventi e gli id dei propri amici.
4. Il server di google per le richieste GCM

Le comunicazioni avvengono tramite richieste http.

L'applicazione è stata progettata pensando agli smartphone e meno ai tablet (nulla vieta l'utilizzo dell'applicazione sul tablet, anche se un'interfaccia più adatta al dispositivo sarebbe opportuna).

L'applicazione è stata scritta per la versione 4.0 per l'utilizzo di alcune componenti introdotte in tale versione, non estremamente necessarie e che sono utilizzabili su versioni precedenti importando delle librerie esterne. Per utilizzare l'applicazione è necessaria una connessione a internet, ed eventualmente un Gps per ottenere informazioni più precise a riguardo della propria posizione.

Implementation:

Le funzioni principali del sistema sono state implementate:

1. Login/ Logout
2. Visualizzazione degli eventi sulla mappa
3. Filtro della mappa
4. Invio richieste di amicizia /conferma richieste d'amicizia
5. Invio richieste ai propri contatti in rubrica che utilizzano l'applicazione.
6. Invio richieste di amicizia ai propri amici di facebook che utilizzano l'applicazione
7. Creazioni degli eventi dal terminale
8. Visualizzazione degli eventi
9. Notifiche GCM
10. Aggiornamento delle informazioni sul terminale da quelle del server
11. Ottenere i propri eventi da face book
12. Aggiunta dell'evento su google calendar
13. Centrare l'evento sulla mappa
14. Chiamata il navigatore per ottenere le informazioni stradali per raggiungere il luogo.
15. Informare i propri amici di un evento.

Per implementare il progetto è stato utilizzato eclipse sia per la parte android che per la parte server in php. Sono state utilizzate se librerie di google per la gestione della mappa e le librerie di facebook. Le parti più interessanti da un punto di vista implementavo sono le richieste per ottenere le informazioni da face book e quelle per selezionare i contatti che possono utilizzare l'applicazione. La parte più difficile da implementare è stata quella relativa ai fragment della activity principale dove sono presenti gli eventi gli

amici e le mappe.

Evaluation:

Il sistema è stato testato principalmente su uno smartphone reale, anche se qualche breve test è stato compiuto sul simulatore. Per il tipo di applicazione creata sarebbero necessari numerosi test con molti utenti per verificare quante risorse server sono necessarie. Per mancanza di tempo (e di fondi) il test è stato affidato a una sola persona esterna che ha evidenziato alcuni problemi. Alcuni relativi al server, che spesso cade o ha tempi di risposta abbastanza lunghi. Altri relativi al basso numero di eventi disponibili e alle opzioni di scelta. Dal punto di vista dei bug non sono stati rilevati bug che compromettono l'usabilità del sistema. Da un punto di vista dell'idea è stato effettuato un sondaggio chiedendo ad alcune persone cosa ne pensavano dell'applicazione e 10 persone su 18 delle intervistate sarebbero interessati ad utilizzare l'applicazione. Gli altri sono abbastanza scettici sulla veridicità degli eventi pubblici disponibili mentre per gli eventi privati trovano che sia più comodo organizzarsi a voce.

Sono state tentate delle ottimizzazioni su quando e quanto spesso è necessario aggiornare le informazioni sul database dello smartphone. Arrivando alla scelta di aggiornare quando arriva una notifica gcm o manualmente.

Limitations:

Il problema maggiore di quest'applicazione è la componente server necessaria, e che per gli eventi pubblici è difficile creare (ma non impossibile) un sistema di gestione per verificare l'attendibilità delle informazioni se non si conosce chi ha creato l'evento. Una possibile soluzione sarebbe quella dotare gli utenti di valutazione per valutare gli eventi e i gli altri utenti, in modo tale da minimizzare scherzi o problemi agli utenti che fanno affidamento sulla veridicità degli eventi pubblici.

Team Members' Contribution:

Essendo da solo ho implementato sia la parte client che la parte server. Per le classi implementate all'interno delle classi Android sono intorno alle 7600 righe di codice compresi gli import e le i commenti, che si possono arrotondare per il codice reale sulle 4500 righe di codice puro. La parte server è in php e le righe di codice puro implementato sono sulle 800 righe. Oltre a questo sono presenti altre 800 righe di codice per i documenti xml.

Lessons learned: Se dovessi implementare di nuovo l'applicazione inizierei con una progettazione migliore pensando alla scalabilità e all'utilizzo di interfacce in modo tale da rendere l'applicazione più modulare possibile, cercando di ottimizzare lo scambio di risorse e l'aggiornamento delle informazioni.

References:

Le informazioni sono state recuperate da:

Android: Guida per lo sviluppatore di Massimo De Carli

Per i vari problemi è stato utilizzato stackoverflow:

<http://stackoverflow.com/questions/tagged/android>

Per le nuove notifiche google cloud messaging:

<http://developer.android.com/guide/google/gcm/index.html>

Facebook:

<http://blog.doityourselfandroid.com/2011/02/28/30-minute-guide-integrating-facebook-android-application/>

<https://developers.facebook.com/tools/>

<http://ericosgood.com/prog/facebook-android-sdk-tutorial/>

Google maps:

<http://mobiforge.com/developing/story/using-google-maps-android>

Varie:

<http://www.androidhive.info/>

Code appendix: il codice è diviso in due file principali la parte server in php e la parte client per android
Per le classi java il codice è stato diviso in 7 package:

1. it.unitn.letsgo contiene tutte le activity e le classi principali del progetto
2. it.unitn.letsgo.adapter contiene gli adapter per le liste
3. it.unitn.letsgo.conessione contiene le classi per il parsing del json e la comunicazione col server
4. it.unitn.letsgo.database contiene le classi per l'uso del database
5. it.unitn.letsgo.fragment contiene le classi che implementano i fragment per la mainActivity
6. it.unitn.letsgo.picker contiene le classi per i picker per la selezione dei dati
7. it.unitn.letsgo.varie contiene quello che non era presente nelle altri package come le classi per incapsulare le informazioni le richieste a facebook alcune variabili condivise a più classi

oltre a questi sono presenti le risorse e i file del server in php

Note: per testare l'applicazione è necessario far partire il server in php impostando sulle variabili condivise l'indirizzo ip del server.