

ALMA MATER STUDIORUM • UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Un Sistema Location – Based per la Mappatura degli Access Point su Android

Relatore:
Dott. Vittorio Ghini

Candidato:
Alberto Scarlino

Sessione II
Anno Accademico 2013/2014

INDICE

| | |
|--|-----------|
| 1. INTRODUZIONE | 4 |
| 2. LE BASI DELLA PROGETTAZIONE | 6 |
| 2.1 Android..... | 6 |
| 2.2 Global Position System..... | 8 |
| 2.3 La rete WI-FI..... | 11 |
| 3. OBIETTIVI | 14 |
| 3.1 Obbiettivi di miglioramento..... | 14 |
| 3.2 Porting per Versione di Android precedenti..... | 14 |
| 3.3 Mappe in modalità offline..... | 16 |
| 3.4 Implementazione di un Database online..... | 17 |
| 4. STRUMENTI | 19 |
| 4.1 Dispositivi Mobili..... | 19 |
| 4.2 Software Development Kit (<i>SDK</i>)..... | 20 |
| 4.3 Google Maps per Android..... | 22 |
| 4.4 L.A.M.P. (<i>Linux</i> , <i>Apache</i> , <i>MySQL</i> , <i>PHP</i>)..... | 23 |
| 4.5 Application Programming Interface..... | 24 |
| 4.5.1 API Key Google Maps..... | 25 |

| | |
|--|-----------|
| 5. IMPLEMENTAZIONE DEL PROGETTO | 27 |
| 5.1 Panoramica..... | 27 |
| 5.2 Architettura dell'applicazione..... | 27 |
| 5.3 Interfaccia..... | 28 |
| 5.4 Android Manifest..... | 29 |
| 5.5 Mappe..... | 30 |
| | |
| 6. IL DATABASE ONLINE | 32 |
| 6.1 Metodi di Visualizzazione..... | 32 |
| 6.2 Acquisizione della Posizione..... | 36 |
| 6.3 Connettività verso la Rete..... | 36 |
| 6.4 Creazione/Aggiornamento tabella Reti..... | 38 |
| 6.5 Creazione/Aggiornamento tabella Punti..... | 40 |
| 6.6 Creazione/Aggiornamento tabella Users..... | 43 |
| 6.7 Classe JSON Parser..... | 47 |
| | |
| 7. SPERIMENTAZIONE | 49 |
| 7.1 Smartphone..... | 49 |
| 7.1.1 Il primo Smartphone Android..... | 50 |
| 7.2 Caratteristiche Dispositivi Usati..... | 51 |
| 7.3 Risultati ottenuti..... | 52 |
| 7.4 Test su HTC Wildfire..... | 53 |
| | |
| CONCLUSIONI E SVILUPPI FUTURI | 61 |
| | |
| BIBLIOGRAFIA | 63 |

1. INTRODUZIONE

Il sistema operativo più diffuso, per dispositivi mobili, è Android. Creato da Google Inc. è presente sull'83,6% dei 320 milioni di smartphone venduti globalmente nel terzo trimestre del 2014.

I ricercatori di Strategy Analytics hanno certificato il successo dell'androide. Con un market share in aumento di due punti rispetto all'81,4% del 2013.

Android è un sistema che adotta una politica di licenza open source. Per questo, il suo codice sorgente, può essere modificato e distribuito liberamente. Inoltre, dispone di un gran numero di sviluppatori che hanno realizzato oltre 700.000 applicazioni con l'obiettivo di aumentare le funzionalità dei dispositivi.

Anche noi, come i tanti sviluppatori di applicazioni Android, abbiamo contribuito nel miglioramento di un'applicazione per la mappatura degli Access Point.

Il lavoro è stato svolto in collaborazione con Manuela Corina basandoci sul lavoro implementato da altri colleghi.

WifiMapping è un'applicazione che ha l'obiettivo di mappare i punti, mediante l'utilizzo del GPS (*Global Positioning System*), di una rete Wi-Fi (*sfruttando le API di Google Maps*) a cui un utente si connette.

Nel 2014 la rete Wi-Fi è molto diffusa in aeroporti, università, stazioni, ristoranti, centri commerciali e altri posti comuni. Permette ai terminali di collegarsi tra loro in maniera wireless (senza fili) utilizzando il protocollo standard IEEE 802.11.

Per questo motivo si è pensato di sviluppare un'applicazione in grado di memorizzare e mappare tutti i punti dove è possibile collegarsi, in modo gratuito, ad internet

usando il nostro Smartphone.

Per tenere traccia di questi punti si utilizza un meccanismo in grado di determinare la posizione geografica del dispositivo mediante il quale l'utente si connette alla rete.

Il GPS è in grado di fornire, ad un dispositivo mobile o ricevitore GPS, le coordinate geografiche in ogni punto della terra o nelle immediate vicinanze ove non ci siano ostacoli per il ricevitore.

L'applicazione è stata sviluppata per smartphone che eseguono sistemi operativi Android 2.2 Froyo in poi.

Verranno descritti tutti gli sviluppi che hanno portato a migliorare questa applicazione.

Dalla porting di versione alla possibilità di scaricare le mappe direttamente sul dispositivo e inoltre la possibilità di condividere, con tutti gli utenti che usano l'applicazione, un database online con tutti i punti registrati delle reti Wi-Fi libere.

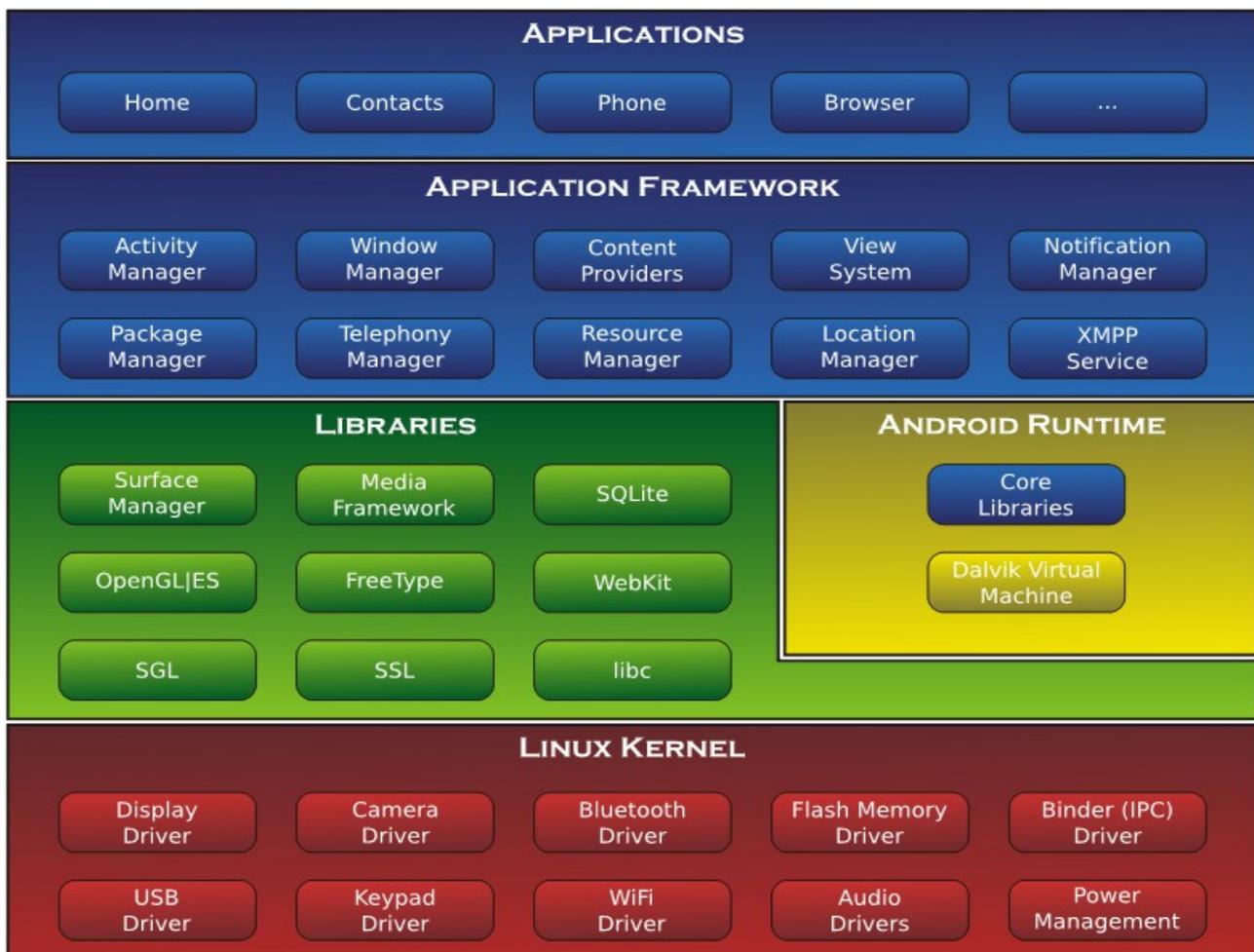
Infine si relazionano i possibili futuri miglioramenti dell'applicazione.

2. LE BASI DELLA PROGETTAZIONE

2.1 ANDROID

Android nasce nell'ottobre del 2003, inventato da Andy Rubin, padre fondatore del progetto. Android è il sistema operativo mobile più diffuso al mondo. Basato sul Kernel Linux è composto da applicazioni Java che vengono eseguite su framework , basato anch'esso su Java e orientato agli oggetti, a sua volta eseguito su un nucleo costituito da librerie Java eseguite tramite la macchina virtuale Dalvik dotata di un compilatore JIT (just-in-time). Inoltre il kernel di android ha un sistema middleware per le comunicazioni, cioè un software di comunicazione che permette di far interagire applicazioni e componenti software.

Le librerie sono porzioni di codice che vengono caricate dal kernel in modo da evitare che numerosi programmi, con le stesse funzioni, abbiano le stesse sezioni di codice replicate in memoria centrale e quando si vuole fare uso di esse, basta solo richiederle. Molte sono le componenti che compongono queste librerie sviluppate in linguaggio C: un surface manager, un framework multimediale OpenCore, un sistema a database relazionali SQLite, API grafiche 3D OpenGL ES 2.0, un motore grafico e di layout basato su WebKit, un motore grafico SGL, SSL ed una libreria libc Bionic, libreria standard per C basata in gran parte su BSD.



Schema di architettura Android

La caratteristica più evidente di Android, come vedremo in seguito in dettaglio, è che le sue diverse versioni sono indicate con un numero di versione secondo gli standard informatici, ma che alla fine queste vengono di preferenza distinte per il proprio “*codename*“, tradizionalmente ispirato alla pasticceria e rigorosamente in ordine alfabetico:

- **Cupcake** (Android 1.5)
- **Donut** (1.6)
- **Eclair** (2.0/2.1)
- **Froyo** (2.2)

- **Gingerbread** (2.3/2.4)
- **Honeycomb** (3.0/3.1/3.2)
- **Ice Cream Sandwich** (4.0)
- **Jelly Bean** (4.1/4.2/4.3)
- **KitKat** (4.4)
- **Lollipop** (5.0)

La presentazione ufficiale del sistema operativo android avviene il 5 novembre del 2007. Qualche giorno dopo viene reso disponibile il primo Software Development Kit (*SDK*) per gli sviluppatori che include: gli strumenti di sviluppo, le librerie, un emulatore del dispositivo, la documentazione, alcuni progetti di esempio, tutorial e altro.

All'inizio Android presentava un numero molto ridotto di applicazioni: un browser basato su webkit, una rubrica e un calendario sincronizzati con Gmail e poco altro. Per invogliare i programmatori a cimentarsi con questa nuova piattaforma, Google istituì nel gennaio del 2008 un concorso con un montepremi di 10 milioni di dollari per le migliori 50 applicazioni.

2.2 GLOBAL POSITION SYSTEM

Il GPS (*Global Position System*) è un sistema costituito da una rete di satelliti che orbitano nello spazio ad una distanza ben precisa dalla terra, inviando segnali ai ricevitori GPS. Oltre a questi segnali sono associati: un codice temporale e dati geografici, che consentono agli utenti di tutto il pianeta di stabilire con esattezza la propria posizione, l'ora e la velocità.

Tale sistema è stato originariamente progettato come strumento militare e di intelligence negli USA.

Transit è stato il primo sistema inventato e collaudato dalla Marina Militare nel 1960. Era composto da cinque satelliti in grado di stabilire le coordinate geografiche delle navi.

Nei successivi anni il GPS conobbe un forte sviluppo non solo in ambito militare ma anche negli usi civili e commerciali sviluppando un'enorme crescita a partire dai navigatori sui veicoli alle più recenti implementazioni sugli Smartphone di ultima generazione.



Funzionamento del GPS

Esistono vari metodi per geolocalizzare, tra i più noti si cita l'analisi dell'indirizzo IP del dispositivo, la triangolazione è basata sulla cella telefonica nella quale il dispositivo si trova e l'utilizzo di un ricevitore GPS. Alcuni di questi metodi sono disponibili anche per computer, mentre altri si trovano per lo più solamente nei dispositivi mobili.

L'individuazione della posizione tramite indirizzo IP è possibile per ogni dispositivo connesso alla rete Internet, occorre “semplicemente” utilizzare un servizio che si occupa di associare l'indirizzo IP del dispositivo ad una coppia di coordinate.

Per ottenere un posizionamento più preciso, sono stati integrati dei servizi chiamati *LBS (Location Based Services)*.

Un terminale GPS, però, acquisisce con lentezza la propria posizione al momento dell'accensione, dovuta alla necessità di ricercare i satelliti in vista, ed il conseguente notevole impegno di risorse hardware ed energetiche.

A tal punto, si è ritenuto opportuno introdurre il sistema **Assited GPS** (*A-GPS*) ricavando in pochi secondi la propria posizione iniziale, in quanto si assume che i satelliti in vista dalla cella siano gli stessi visibili dai terminali sotto la sua copertura radio.

Il GPS però presenta molti problemi se utilizzato in ambienti chiusi oppure in città, tra gli edifici. Per questo sono state pensate alcune soluzioni alternative: una di queste è l'utilizzo del segnale *GSM* o *UMTS* (*Universal Mobile Telecommunication System*), tecnologia di telefonia mobile di terza generazione (*3G*) successore del GSM, ricevuto dalle celle circostanti che si può estendere di alcuni km in zone rurali.

Esistono in commercio ricevitori GPS (*esterni*), interfacciabili mediante porta USB o connessioni senza fili come il Bluetooth, che consentono di realizzare navigatori GPS su vari dispositivi: palmari, PC, computer portatili, e, se dotati di sufficiente memoria, anche telefoni cellulari.

Per la navigazione esistono software appositi, proprietari o open source che utilizzano una cartografia, anch'essa pubblica o proprietaria.

2.3 LA RETE WI-FI

Il Wi-Fi (*wireless*) è nato tra la metà degli anni '80 e l'inizio degli anni '90.

È una tecnologia inventata per lo scambio di dati tra due o più dispositivi senza fili utilizzando le onde radio. Questo sistema di comunicazione permette la condivisione non solo di file ma anche di una connessione ad internet basandosi sulle specifiche dello standard IEEE 802.11.

Ogni rete può essere composta da uno o più Access Point (*o hotspot*), ovvero un punto d'accesso o più punti d'accesso alla rete che fungono da sorgente del segnale, e uno o più clienti che si connettono alla rete.

L'area che può coprire un segnale wireless è tra i 50 ed i 100 metri (questo dipende dall'area architettonica), ma può essere esteso in diversi modi. Si può ampliare attraverso il collegamento di differenti access point tramite cavo; oppure creando un ponte wireless con ripetitori di segnali come gli amplificatori Wi-Fi che catturano il segnale di un access point e lo ritrasmettono.

Il funzionamento di un access point (solitamente un router dotato di una o più antenne Wi-Fi) è semplice. Trasmette ogni 100 ms un pacchetto dati chiamato beacon contenente lo SSID (*Service Set Identifier*) della rete e altre informazioni. Il cliente (un computer o dispositivo mobile dotato di scheda Wi-Fi o un ripetitore di segnale) può decidere di connettersi alla rete seguendo vari criteri: collegarsi alla rete con un SSID noto (cioè già connesso in precedenza) oppure quella del segnale più forte e che quindi garantisce le prestazioni migliori.

Accessi Wi-Fi sono ormai disponibili in aeroporti, stazioni ferroviarie, internet point sparsi per il mondo. In Europa è diffusa la rete dei "*Totem Freestation*".



Wi-Fi Zone

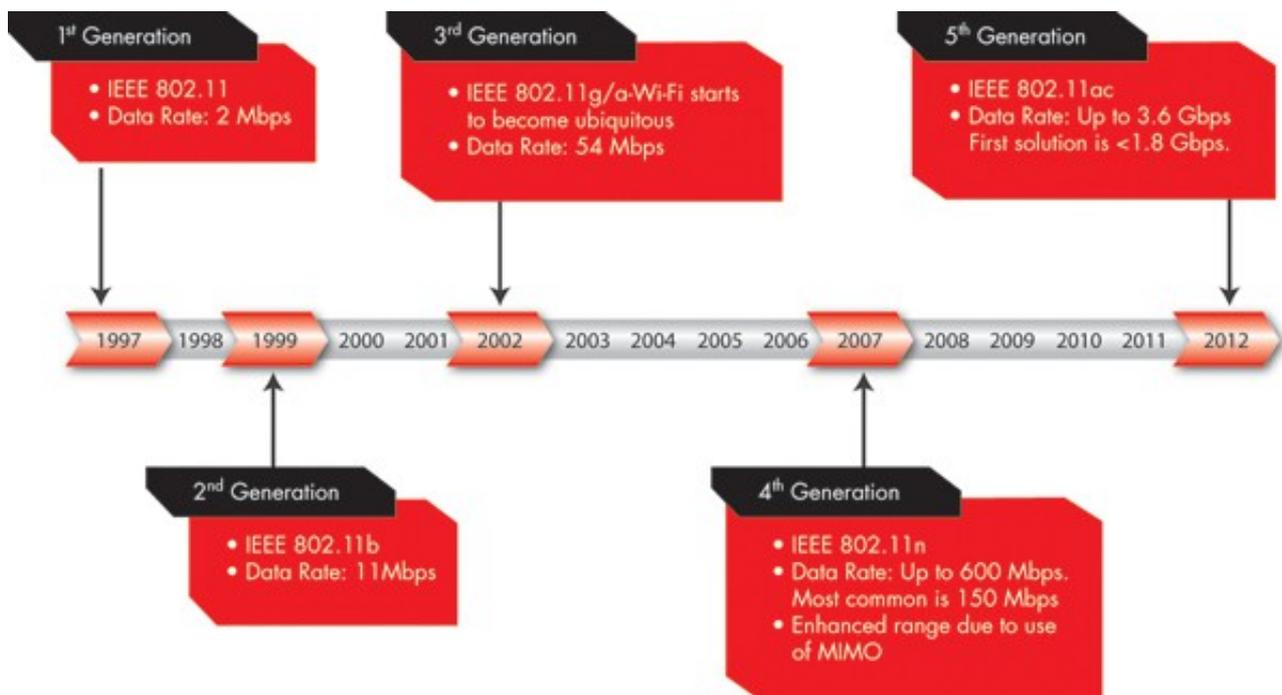
Un problema nelle reti wireless è la necessità di misure di sicurezza contro l'accesso abusivo alla rete e l'intercettazione dei dati in transito. Mentre in una realtà di rete cablata la sicurezza del mezzo è racchiusa "tra le proprie mura" (e quindi relativamente più semplice gestire il controllo degli accessi al mezzo fisico), in un sistema di reti wireless il "mezzo fisico" è l'aria, per cui un malintenzionato potrebbe ottenere l'accesso alla rete senza fisicamente introdursi all'interno delle mura.

Col tempo si sono sviluppati diversi sistemi di sicurezza. Una delle tecniche più semplici è quella di consentire l'accesso al proprio Access Point solo a certi *MAC* Address. Dato che i *MAC* address possono essere facilmente clonati, è diventato necessario introdurre sistemi di sicurezza più efficaci.

La maggior parte degli Access Point implementano un sistema di cifratura dei dati denominato *Wired Equivalent Privacy* (WEP), ma anche questo sistema si è rivelato con il tempo poco affidabile (il segnale è stato decodificato in pochi minuti con una crittoanalisi).

I più recenti Access Point implementano sistemi di crittografia denominati *Wi-Fi Protected Access* (WPA e WPA2), fornendo un più avanzato metodo di sicurezza.

Il più sicuro è “WPA” poiché permette di crittografare le chiavi di sicurezza con l'algoritmo AES (*Advanced encryption Standard*) a 256 bit.



Wi-Fi Timeline

3. OBIETTIVI

3.1 OBIETTIVI DI MIGLIORAMENTO

L'obiettivo di questo progetto è quello di migliorare il prototipo esistente di un'applicazione “*WiFiMapping*” nei seguenti punti:

- porting per versioni di Android precedenti alla 4.0
- possibilità di utilizzare delle mappe in modalità offline: questo renderebbe l'applicazione più utile dal momento che attualmente è possibile utilizzare le mappe di Google solo se si è connessi ad una rete
- sviluppo di un database online da affiancare a quello locale: in tal modo gli utenti potrebbero condividere le proprie registrazioni (solo quelle che non richiedono una password e quindi utilizzabili da tutti).

3.2 PORTING PER VERSIONI DI ANDROID PRECEDENTI

L'applicazione, in origine, è stata sviluppata per Smartphone che eseguono sistemi operativi Android ed in particolare dalla versione Android 4.0 in poi.

Attualmente è stata realizzata la porting per versione precedenti di Android 2.2.

Ciò è avvenuto modificando AndroidManifest.xml in cui, con il seguente codice:

```
<uses-sdk android:minSdkVersion="8"
          android:targetSdkVersion="17"/>
```

Si include la minima versione 2.2 (con API 8) alla versione di Android 4.2.2 (con API 17).

Si è effettuato, successivamente, l'aggiornamento dell'uso delle Google Maps Android v1 API per il nostro terminale.

Attualmente le Google Maps Android v1 API sono state disapprovate pur continuando a usufruirne di esse, ma si è già a disposizione di una nuova versione v3.

Queste consentono di manipolare i dati delle mappe. La libreria Maps è esterna alla libreria standard di Android fornito nel SDK.

Per ottenere questo è stato necessario installare:

- Google API add-on (estensioni di Android SDK)
- Riconfigurare un progetto esistente su cui applicare le Google API add-on
- Configurare Android Virtual Device
- Aggiunta di elementi in uses-libreria nel Manifest dell'applicazione in uso:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.example.package.name">
    ...
    <application android:name="MyApplication" >
        <uses-library android:name="com.google.android.maps" />
        ...
    </application>
    ...
</manifest>
```

- Aggiunta di Google Maps Android API v1 key all'applicazione per fornire i servizi offerti.
- Firmare l'applicazione attraverso l'uso del certificato API Key

3.3 MAPPE IN MODALITA' OFFLINE

Questo miglioramento potrebbe dare la possibilità di utilizzare le mappe in modalità offline rendendo l'applicazione più utile nel momento in cui non si è connessi ad una rete.

Inizialmente, ciò non era possibile in quanto vi era violazione dei diritti di privacy di Google Maps.

Ma negli ultimi mesi, sembrano esserci enormi sviluppi in quanto, Google Maps consente il download delle mappe offline, senza appunto utilizzare una connessione dati, utile nel momento in cui si è lontani da casa o ufficio e non si dispone di una connessione dati.



Mappa Dipartimento di Informatica

È possibile scaricare preventivamente le mappe Google Maps relative alle aree geografiche interessate e quando si ha bisogno di consultare la mappa in modalità offline.

Google Maps non dovrà scaricare i dati già scaricati sullo Smartphone, ma semplicemente richiede al server di Google le informazioni per la corretta individuazione e visualizzazione di indirizzi stradali.

Usare Google Maps offline è semplice. Bisogna verificare di aver installato, sul dispositivo Android, l'ultima versione dell'applicazione Google Maps, qualora non fosse aggiornata la si aggiorna tramite Google Play. E' necessario essere connessi ad una rete per il download dei propri dati per poterli poi consultare nel momento in cui ce ne sia il bisogno.

3.4 IMPLEMENTAZIONE DI UN DATABASE ONLINE

I dispositivi Android integrano un DBMS (*Database Management System*) che proviene dal mondo dell'Open Source.

Si è pensato di sviluppare un database online da affiancare a quello locale, in tal modo gli utenti potrebbero condividere le proprie registrazioni (solo quelle che non richiedono una password e quindi utilizzabili da tutti).

È stato usato **LAMP**, acronimo di **Linux, Apache, MySQL e PHP**. Innanzitutto si è creato un database utilizzando **PhpMyAdmin**.

CREATE DATABASE Mapping;

PhpMyAdmin non è altro che un'interfaccia grafica che permette di amministrare **MySql**, un tipo di database che immagazzina qualsiasi tipo di dati in strutture chiamate tabelle.

In pratica, possiamo visualizzare il contenuto del nostro database; creare, modificare, cancellare intere tabelle o singoli record; fare un backup dei dati contenuti; visualizzare

informazioni interessanti sul database.

Il database, simile a quello locale, è composto da 3 tabelle: Reti, Punti e Users che verranno illustrati nel capitolo 6.

4. STRUMENTI

4.1 DISPOSITIVI MOBILI

Al giorno d'oggi, i dispositivi mobili, rivestono un ruolo molto importante nella nostra vita quotidiana, permettendoci di compiere operazioni e compiti che, fino a qualche anno fa, erano eseguibili solo attraverso un normale PC.

L'evoluzione tecnologica, con il passare del tempo, ha potenziato lo sviluppo dei normali PC ma ha coinvolto i dispositivi così detti “mobili”. Evoluzione che li ha trasformati da semplici organizer “da tasca” a veri e propri terminali ricchi di funzionalità, discreta potenza di calcolo ma soprattutto di connettività.

La connettività, nei dispositivi mobili, ha avuto risultati sbalorditivi il campo della telefonia cellulare i cui sistemi si sono evoluti in:

1. ***Sistemi di Prima Generazione:***

sistemi cellulari analogici, alle frequenze di 450 e 900 Mhz, che permettono il roaming solo a livello nazionale.

Abbiamo lo standard **TACS** (*Total Access Communication System*) che garantisce una buona qualità della voce ed un elevato grado di copertura del territorio nazionale.

2. ***Sistemi di Seconda Generazione:***

Con l'arrivo della seconda generazione c'è stato il passaggio dall'analogico verso il digitale. Lo standard più utilizzato è stato inizialmente il **GSM** (*Global System for Mobile*

Communications), che permette il trasferimento di voce e dati digitali di volume ridotto. Seguito poi dal GPRS (*General Packet Radio System*) che permette il trasferimento di voce e dati digitali di medio volume e dal EDGE (*Enhanced Data Rates for Global Evolution*) migliorando di gran lunga la capacità di banda del GPRS.

3. ***Sistemi di Terza Generazione:***

In ultimo abbiamo i sistemi di terza generazione 3G in cui troviamo la compatibilità dei servizi mobili di terza generazione con le reti di seconda generazione grazie alla tecnologia UMTS (*Universal Mobile Telecommunications System*) che permette il trasferimento simultaneo di voce e dati digitali ad alta capacità di banda. Seguito in ultimo dal HSDPA (*High-Speed Downlink Packet Access*).

4.2 **SOFTWARE DEVELOPMENT KIT (SDK)**

Il Software Development Kit (SDK) , vale a dire un kit di sviluppo di software.

Attraverso questo kit di sviluppo, qualsiasi programmatore potrà procurarsi il codice sorgente di ogni versione Android e modificarlo o migliorarlo autonomamente.

Un passo indietro: il codice sorgente è un testo scritto in linguaggio di programmazione che, una volta elaborato, porterà alla realizzazione di un programma eseguibile dal processore.

I linguaggi di programmazione alla base della maggior parte delle applicazioni per Android sono Java e C/C++.

Con l'SDK, scaricabile dal sito ufficiale, sarà quindi possibile disporre su PC di un'interfaccia per lanciare applicazioni Android e modificarle.

Le applicazioni Android non pongono alcuna differenza tra le applicazioni di base del dispositivo e le applicazioni di terze parti: possono essere costruite per avere pari accesso

alle funzionalità del dispositivo.

Android consente di accedere ad una vasta gamma di librerie e strumenti utili per lo sviluppo delle applicazioni. Quest'ultime si compongono di quattro mattoni fondamentali:

1. **Activity:** (le attività): sono la componente fondamentale delle applicazioni Android. Un'attività è una classe che estende *android.app.Activity*. L'autore del codice, realizzando l'attività, si serve dei metodi ereditati da Activity per controllare cosa appare nel display, per assorbire gli input dell'utente, per intercettare i cambi di stato e per interagire con il sistema sottostante.

In questa applicazione l'Activity principale è *MapsActivity.class* che permette all'utente di visualizzare la mappa Google e di interagire con essa. Permette inoltre di accedere ad un menu di opzioni che servono ad attivare funzionalità aggiuntive (tra cui la modalità di visualizzazione).

2. **Broadcast Receiver:** viene utilizzato quando s'intende intercettare un particolare evento attraverso tutto il sistema.

La classe da estendere è *android.content.BroadcastReceiver*. Questa classe opera in background e mantiene le sue funzionalità anche quando l'applicazione è apparentemente chiusa. In questo caso si occupa di monitorare i cambi di stato del Wi-Fi (connesso, in connessione, disconnesso, in disconnessione, stato sconosciuto) e di notificarli all'utente mediante un messaggio temporaneo (*Toast.class*).

3. **Location Listener:** si tratta di un oggetto, di tipo *android.location.LocationListener*, su cui vengono notificate tutte le misurazioni effettuate e filtrate secondo alcuni parametri.

Nell'applicazione che si sta creando, il *Listener* viene attivato ogni qual volta l'interfaccia GPS è attiva. Nel momento in cui si ci collega ad una rete Wi-Fi, il Location Listener si occupa di salvare i punti e le informazioni relative alla rete nel Database.

4. **Overlay:** per disegnare elementi su una mappa Google, bisogna

sovrapporre ad essa degli “overlay”, ognuno dei quali mostra qualcosa in particolare. Per realizzare un overlay si deve estendere la classe *com.google.android.maps.Overlay*. In tal caso le classi *MyOverlay.class* e *Itemized.class* si occupano di disegnare sulla mappa i punti registrati nel Database e per ciascuno di essi, un marker che se cliccato fornisce all'utente le informazioni sulla rete.

Per quanto riguarda l'installazione di Android SDK, il procedimento è alquanto semplice. Una volta scaricato il kit di sviluppo dal sito ufficiale, si dovrà decomprimere il file in una cartella a scelta, lanciare SDK Manager, che verificherà la presenza di eventuali pacchetti di sviluppo installati sul vostro PC e vi mostrerà la lista dei pacchetti disponibili per il download.

4.3 GOOGLE MAPS PER ANDROID

Google Maps è un servizio messo a disposizione da Google sia via web che per Android. Consente la ricerca e la visualizzazione di mappe geografiche di tutta la Terra.

Viene aggiornato in continuazione introducendo la ricerca di servizi in particolari luoghi, tra cui ristoranti , monumenti, negozi, trovare un possibile percorso stradale tra due punti. Ma la cosa che a noi interessa di più è la visualizzazione di immagini in alta definizione davvero molto dettagliate. Per alcune zone, l'opzione dello zoom ci fa distinguere, addirittura, in molti casi le case, i giardini, le strade e così via.

È essenziale per pianificare viaggi, infatti è sufficiente inserire semplicemente città ed indirizzo di partenza e di destinazione, con la possibilità di scegliere il mezzo di trasporto utilizzato e le eventuali tappe; in questo modo Google Maps non solo fornirà le indicazioni stradali, ma calolerà anche il tempo necessario per arrivare a destinazione e il numero di chilometri da percorrere.

A differenza di Google Maps che è un sito, introduciamo il concetto di Google Earth

che è un software capace di generare immagini virtuali della terra utilizzando immagini satellitari, fotografie aeree e dati topografici della piattaforma GIS (*sistema informatico computerizzato*) con un dettaglio molto elevato.



Google Maps, Google Earth

4.4 L.A.M.P. (LINUX, APACHE, MYSQL, PHP)

Oltre all'uso del SDK, per lo sviluppo e il miglioramento dell'applicazione , si è usato L.A.M.P. per implementare il database online.

L.A.M.P. è una piattaforma di sviluppo di applicazioni web. Il suo acronimo è composto dalle prime lettere delle parole Linux, Apache, MySQL e PHP.

Il termine è apparso per la prima volta nel 1998 su una rivista informatica. L'articolo della rivista dimostrava come fosse possibile, attraverso l'integrazione di alcuni software liberi, realizzare una piattaforma web paragonabile alle soluzioni commerciali di allora.

Se all'inizio si trattava solo di combinare alcuni pacchetti software gratuiti, per ottenere una soluzione in grado di presentare contenuti dinamici via web, con il tempo si è assistito ad una sempre più forte integrazione dei vari componenti.

Attualmente esistono varie distribuzioni GNU/Linux che permettono l'installazione e la configurazione di soluzioni L.A.M.P. in pochi e semplici passi. A volte viene utilizzata in alternativa anche la definizione “*the LAMP stack*”, ovvero la pila Linux, Apache, MySQL e PHP i cui ruoli dei singoli componenti sono facilmente identificabili:

- *Linux* come sistema operativo
- *Apache* come web server
- *Mysql* come RDBMS (*Relational DataBase Management System*)
- *PHP*, Perl e/o Python come linguaggio di scripting che permette l'ottima usabilità e ottima interfaccia web.

4.5 APPLICATION PROGRAMMING INTERFACE

Con Application Programming Interface (in italiano *Interfaccia di Programmazione di un'Applicazione*), si indica ogni insieme di procedure disponibili al programmatore.

La finalità è quella di ottenere un'astrazione a più alto livello, permettendo di espandere le funzionalità di un programma solitamente tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione.

Le API evitano ai programmatori di riscrivere ogni volta tutte le funzioni necessarie al programma da zero, ovvero dal basso livello, rientrando quindi nel concetto di riuso del codice. Le API stesse rappresentano quindi un livello di astrazione intermedio: il software che fornisce una certa API è detto implementazione dell'API.

4.5.1 API KEY GOOGLE MAPS

Per poter accedere ai servizi di Google Maps bisogna ottenere un' API key, cioè una chiave che permette di poter accedere al servizio, univocamente per ogni dominio.

L'API key, attualmente, sono alla versione 3 che si differenzia dalla versione 2.

Le principali novità sono:

- L' API key non serve più, quindi non è necessario richiedere una nuova key per ogni dominio in cui si vuole includere la mappa.
- Le vecchie variabili globali della versione precedente sono state sostituite da un nuovo namespace denominato google.maps.
- È stata adottata una nuova architettura MVC (*Model-View-Controller*), il codice è ancora più modulare e object oriented.
- Sono stati migliorati i tempi di caricamento delle mappe grazie alla riduzione del peso (circa 35 kb) del JavaScript.
- Miglioramenti per i browser di tutti i dispositivi mobili (Android in particolare).
- Con le nuove API, l'interfaccia di controllo della mappa, è attiva automaticamente quindi non c'è bisogno di aggiungere i controllo di zoom o di tipo di mappa.

API Project ▾

- Overview
- Services
- Team
- API Access
- Reports
- Quotas

Client ID for installed applications

| | | |
|---------------------------------|---|----------------------------------|
| Client ID: | 48426601409-toeeggbf0gug0trmktk2e8vpps0anoou.apps.googleusercontent.com | Edit settings... |
| Redirect URIs: | urn:ietf:wg:oauth:2.0:oob http://localhost | Download JSON |
| Application type: | Android | Delete... |
| Package name: | sm.hello.example | |
| Certificate fingerprint (SHA1): | 33:4B:BF:45:32:65:5C:CC:15:BF:FB:28:EA:EF:5B:44:9D:2A:C6:7F | |
| Deep Linking: | Disabled | |

[Create another client ID...](#)

Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

| | | |
|---|--|--|
| Key for Android apps (with certificates) | | Generate new key... |
| API key: | AIzaSyDfyzbAtkfsYOnA4SHP67NyyhYgtdd5M1k | Edit allowed Android apps... |
| Android apps: | 33:4B:BF:45:32:65:5C:CC:15:BF:FB:28:EA:EF:5B:44:9D:2A:C6:7F;sm.hello.example | Delete key... |
| Activated on: | Mar 20, 2013 4:22 AM | |
| Activated by: | alberto.scarlino@gmail.com - you | |

| | | |
|---|---|--|
| Key for browser apps (with referers) | | Generate new key... |
| API key: | AIzaSyAQaIFc39xdUkQB4yYjkYx_1-LR1zN--FM | Edit allowed referers... |
| Referers: | Any referer allowed | Delete key... |
| Activated on: | Mar 20, 2013 3:51 AM | |

Generazione API Key

5. IMPLEMENTAZIONE DEL PROGETTO

5.1 PANORAMICA

Un' applicazione Android è costituita da diversi tipi di componenti che comprendono : codice sorgente Java, documenti XML, icone ed immagini, basi di dati e file binari di altro tipo. Tutti questi componenti vanno in un qualche modo assemblati fra di loro ed in base alla tipologia va inserito in un particolare punto del progetto che provvederemo ad elencare.

5.2 ARCHITETTURA DELL'APPLICAZIONE

L'architettura dell'applicazione è composta essenzialmente da un Database e da una serie di metodi che si occupano di:

- Salvare le coordinate del punto in cui ci si trova associandone la potenza del segnale Wi-Fi;
- Disegnare sulla mappa Google tutti i punti salvati nel Database;
- Aggiornare i valori di un punto già memorizzato;
- Poter scegliere tra due modalità di visualizzazione dei punti: *“Wi-Fi Registrati”*

(tutte le reti registrate) e “Potenza Wi-Fi connesso” (potenza del segnale della rete a cui si è connessi);

- Suggestire il punto di connessione migliore tenendo conto della distanza e della potenza, se non si è connessi a nessuna rete.

Ciò è stato effettuato usando la piattaforma L.A.M.P e Eclipse. Quest'ultimo è un ambiente integrato di sviluppo e può essere usato per produrre o avviare un progetto e compilare un semplice programma. Si compone di un Workspace che ha l'obiettivo di memorizzare tutto il codice relativo a un programma o sistema software e un Project che rappresenta dunque una porzione di un sistema software, ovvero una porzione di workspace in pratica, un insieme di classi. Inoltre è possibile usare o modificare diversi plug-in cioè delle componenti software ideate per uno specifico scopo.

Il tutto è stato sviluppato in Eclipse e simulato, grazie all'emulatore di Android presente nel SDK, per controllare i vari aspetti relativi alla corretta esecuzione del software.

5.3 INTERFACCIA

Prima di iniziare a scrivere righe di codice, bisogna impostare alcune variabili necessarie al software per poter sfruttare le risorse della piattaforma.

Le interfacce utente in Android si possono definire in due forme: mediante file XML o direttamente attraverso codice Java. Generalmente, la forma più raccomandabile e allo stesso tempo più semplice è XML, in quanto aiuta a separare la logica dalla presentazione dell'applicazione. Tuttavia, definire l'interfaccia utente in XML, non è sempre la via più facile visto che potremmo trovarci a dover costruire una interfaccia dinamicamente, a tempo di esecuzione.

Un'Activity visualizza l'interfaccia utente dell'applicazione. Quando si crea un nuovo progetto Android in Eclipse, quest'ultimo creerà per noi un file XML (*main.xml*), che

definisce l'interfaccia utente. Quando l'applicazione viene aperta il metodo *onCreate()* dell'activity carica il file *main.xml* contenente la nostra interfaccia utente.

5.4 ANDROID MANIFEST

AndroidManifest.xml è un File che descrive al dispositivo l'applicazione che dovrà eseguire, non tanto in termini di interfaccia grafica, ma raccogliendo le caratteristiche fondamentali, come il nome, l'icona associata e così via. Questo documento viene generato in automatico dal plugin ADT (*Android Development Tools*) una volta terminata la creazione del progetto.

È posizionato nella cartella principale **root** del progetto. All'interno di questo file sono elencate le *Activity* e i *Service* dell'applicazione, con i permessi che necessita per funzionare correttamente e per determinare come questi interagiscono l'un l'altro e con le altre applicazioni. .

Quando l'applicazione viene installata il gestore dei pacchetti concede o non concede i privilegi a seconda di come li abbiamo configurati nell'*AndroidManifest.xml*.

La radice di ogni *AndroidManifest.xml* è il tag `<manifest>`: questo tag include i nodi che definiscono i componenti dell'applicazione, l'ambiente di sicurezza, e tutto ciò che fa parte dell'applicazione. La struttura di Android manifest è costituita nel seguente modo.

Al suo interno troviamo:

- Dichiarazione delle activity che compongono l'applicazione
- Definizione dei permessi che verranno assegnati all'applicazione per l'accesso ad API del dispositivo
- Definizione dei permessi che regolano come una seconda applicazione può utilizzare le componenti della prima
- Il package di appartenenza

- La versione di SDK minima da utilizzare
- La versione dell'applicazione

La root del file xml è il tag “manifest”:

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

5.5 MAPPE

Le mappe utilizzate per la visualizzazione grafica della posizione di ogni utente, sono le Google Maps. Affinchè si possano utilizzare le funzionalità GPS in un'applicazione per Android, bisogna aggiungere dei permessi, in tal caso `access_fine_location`. E altri permessi come `android.permission.INTERNET` per poter navigare con le mappe di Google, oppure l'uso di direttiva `uses-library` per l'utilizzo delle librerie per google maps. Il tutto incluso nel file Android Manifest.

Per poter visualizzare le mappe di Google all'interno dell'applicazione, nel `main.xml` si sostituisce l'elemento `TextView` con l'elemento `MapView`. Ricordiamo che il `main` è il metodo principale dove vengono richiesti tutti i parametri, vengono creati gli oggetti e applicate le modifiche ad essi tramite metodi già definiti.

Attraverso i parametri *latitudine* e *longitudine* che segnalano il cambio di posizione geografica per mezzo di una variazione segnalata dal dispositivo GPS, viene inviata alle Mappe di Google la visualizzazione della posizione corrente sulla mappa.

Abbiamo detto scegliere le Google Maps in quanto la piena compatibilità, e facilità di interazione, tra i prodotti Google è molto efficiente. Infatti nell'SDK di Android, sono presenti delle API, che permettono di rappresentare, in maniera quasi immediata, un punto GPS (*GeoPoint*) sulla Google Map.

6. IL DATABASE ONLINE

I web database sono un particolare tipo di Servizi Web che offrono la possibilità di creare database direttamente sul Web. Questi servizi offrono normalmente tutte le caratteristiche base di un normale database per costruire tabelle strutturate di dati di ogni genere con campi di tipo testo, numeri, data e ora e altri.

Il loro servizio non è solo quello di Repository o contenitore di dati, ma anche quello di creare un'interfaccia grafica usabile per la gestione degli stessi presentandosi come vere e proprie applicazioni web.

6.1 METODI DI VISUALIZZAZIONE

L'applicazione già realizzata precedentemente, mette a disposizione diverse opzioni. Ne presentiamo una di queste che permette di fare uno switch tra due modalità di visualizzazione della mappa: “*Wi-Fi Registrati*” e “*Potenza Wi-Fi Connesso*” .

Con l'opzione “*Wi-Fi Registrati*” verranno visualizzati sulla mappa tutti i punti salvati nella tabella Punti del Database; per ciascun punto verrà disegnata un'area di copertura circolare con raggio di un metro e una volta cliccato fornirà l'SSID della rete e la potenza espressa in RSSI.

Se il GPS è attivo e si è connessi ad una rete Wi- Fi, verranno disegnati sulla mappa

anche i punti che l'applicazione sta registrando al momento, con una frequenza di aggiornamento pari allo spostamento di due metri.

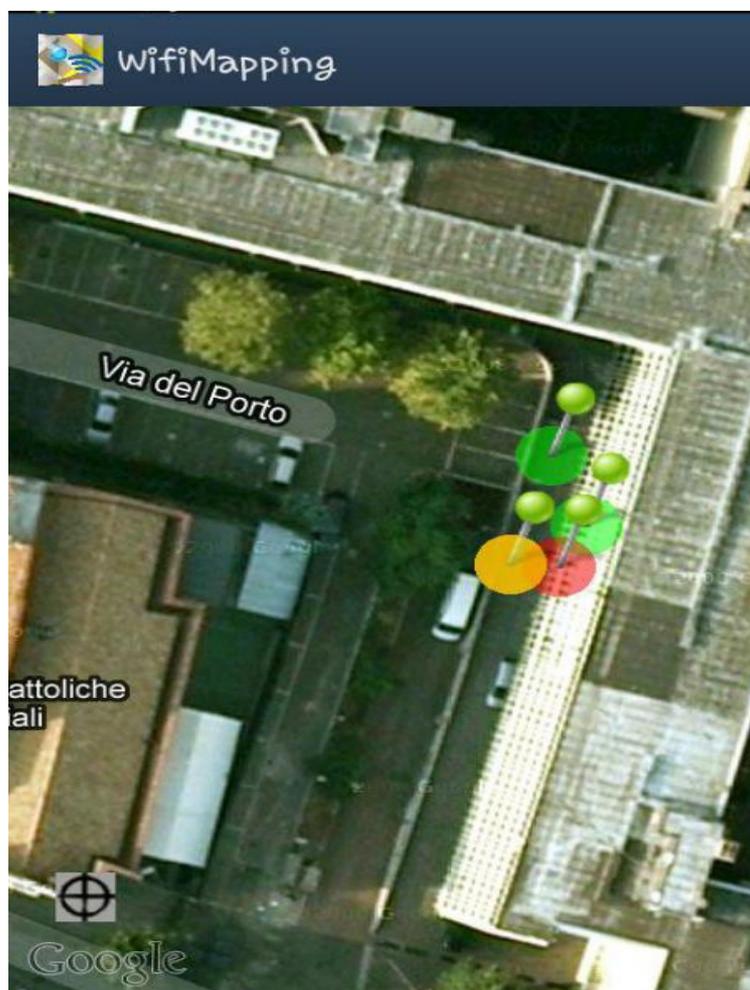
L'applicazione esistente ha provveduto a dare all'utente la possibilità di distinguere facilmente i punti di diverse reti Wi-Fi, semplicemente facendo uso di colori diversi.



Visualizzazione Wi-Fi Registrati

L'opzione “*Potenza Wi-Fi Connesso*” permette di visualizzare sulla mappa tutti i punti salvati nel Database associati alla rete alla quale si è connessi. Se non si è connessi a nessuna rete, la mappa sarà vuota. Per ciascun punto sarà disegnata un'area di copertura circolare con raggio di un metro e una volta cliccato fornirà l'SSID della rete e la potenza espressa in RSSI.

Se il GPS è attivo, verranno disegnati sulla mappa anche i punti che l'applicazione sta registrando al momento con una frequenza di aggiornamento pari allo spostamento di due metri. Anche qui avviene una colorazione delle reti presenti ma diversa da quella precedentemente descritta perché si basa sulla potenza del segnale.



Visualizzazione Potenza Wi-Fi connesso

Nel menù messo a disposizione dall'applicazione, vi è un'altra opzione molto importante: “*Suggerisci Wi-Fi*”.

Tale opzione richiama un metodo che si pone l'obiettivo di dare ad un utente, che ha bisogno di una rete wireless, un suggerimento su qual è quella “migliore” tra le reti presenti nella tabella Reti del Database.

La scelta della rete “migliore” viene effettuata attraverso il seguente algoritmo:

- Verifica della posizione attuale del dispositivo;
- Calcolo della distanza di tale punto con tutti i punti centrali (Middle Point) di tutte le reti conosciute eleggendo la rete più vicina;
- Calcolo della distanza tra il punto in cui ci si trova e ciascun punto della rete più vicina suggerendo quelli con potenza maggiore;
- Nell'eventualità di avere più punti ottimi, viene suggerito quello più vicino.



Menù Funzionalità Aggiuntive

Il punto trovato, quindi, sarà il punto meno distante tra i punti appartenenti alla rete più vicina e con la potenza maggiore. Tale punto verrà suggerito all'utente fornendogli le coordinate (se l'utente dispone di un collegamento UMTS, verrà reindirizzato direttamente all'applicazione "Maps" presente nel sistema operativo Android in modo da poter essere guidato fino al punto suggerito).

Quest'opzione ovviamente può essere visualizzata solo se l'utente non è connesso ad alcuna rete Wi-Fi.

6.2 ACQUISIZIONE DELLA POSIZIONE

Il sistema che si sta sviluppando è costituito da una serie di dispositivi mobili in grado di rilevare la propria posizione (attraverso i parametri longitudine, latitudine) e di scambiarla con altri utenti via rete. Affinchè il dispositivo sia in grado di localizzare se stesso è necessaria un'interfaccia verso un sistema satellitare di posizionamento globale GPS, connesso direttamente al dispositivo. Questa comunica la posizione al dispositivo con una serie di stringhe definite nel protocollo standard.

6.3 CONNETTIVITA' VERSO LA RETE

Nel momento in cui si è acquisita la posizione, il dispositivo sarà in grado di rendere visibile la propria posizione inviandola a tutti gli elementi della rete. Ciò avviene tramite interfacce radio basate su vari protocolli di comunicazione tra i quali i più noti 802.11 (Wi-Fi), UMTS, HSDPA.

Una volta effettuato il tutto, in questo progetto si è reso opportuno aggiungere:

1. La visualizzazione dettagli Rete o Punto

Si sono creati due file php chiamati *details.php* e *details_point.php*. Con questi file si otterranno i singoli dettagli della rete o punto prendendo ID prodotto (PID) come parametro.

Ciò avviene effettuando un metodo GET tramite PHP all'url delle reti presenti nella Tabella Reti. Si attende un tag successo/fallimento. Se successo, visualizza tutte le reti presenti nel database. Se non ci sono reti, viene lanciata un'attività di aggiunta della rete. La stessa cosa avviene per Visualizza Punto. Per prendere tutte le richieste da Java a PHP, si usa JSON.

La risposta *JSON*, per il file di cui sopra, sarà successo rete o punto trovato o fallimento rete o punto non trovato.

2. *La cancellazione di una Rete o Punto*

Si sono creati due file php chiamati *delete.php* e *delete_point.php* per cancellare la rete o punto nel database selezionando l'identificativo PID.

Per Cancellare una rete o un punto si effettua un metodo GET tramite PHP all'url delle reti presenti nella Tabella Reti o Punti. Si attende un tag successo/fallimento.

Se successo permette di cancellare, usando il PID, il record della tabella. Per prendere tutte le richieste da Java a PHP, si usa JSON.

6.4 CREAZIONE/AGGIORNAMENTO TABELLA RETI

La tabella *Reti* si occupa di salvare tutte le reti wifi a cui l'utente si connette tenendo traccia dell'*SSID* (nome della wifi che si identifica agli utenti), del *MAC Address* (indirizzo fisico della rete, un codice di 48 bit univoco assegnato alla scheda wifi) e segnando a ciascuno di essi un id univoco.

| ID | SSID | MAC |
|----|-----------|-------------------|
| 1 | Alice-102 | 06-25-44-2A-4B-F3 |
| 2 | Vodafone | 00-07-64-3F-1D-F9 |
| 3 | AlmaWi-Fi | 12-15-33-6C-2A-D3 |
| 4 | NetGear | 61-11-34-2A-2A-A9 |

```
CREATE TABLE reti (  
    pid int(11) primary key auto_increment,  
    idrete varchar(100) not null,  
    ssid varchar(100) not null,  
    mac text,  
    created_at timestamp default now(),  
    updated_at timestamp  
);
```

Creazione di una Rete

Nel lato server, per creare una rete si utilizza il file php chiamato ***crea_rete.php***. Questo file viene usato principalmente per la creazione di una nuova rete nella relativa tabella.

Ciò avviene prendendo l'ID, SSID e il MAC di una rete, tali parametri vengono passati al database online tramite il metodo POST attendendo come risposta il tag successo/fallimento.

Nel codice si esegue una lettura dei dati della rete tramite il metodo POST e la loro memorizzazione nella relativa tabella. Alla fine si esegue una risposta adeguata ***JSON***.

La risposta sarà: parametro mancante, rete aggiunta alla tabella o errore di inserimento dati.

Aggiornamento di una Rete

Si è creato un file php chiamato ***update.php***. Ogni prodotto è identificato da PID. Per aggiornare una rete si effettua un metodo POST tramite PHP all'url delle reti presenti nella Tabella Reti.

Si attende un tag successo/fallimento. Se successo, permette di modificare la rete

presente nel database.

La risposta **JSON** sarà: successo file aggiornato o fallimento possibile campo mancante.

The screenshot shows the phpMyAdmin interface for a database. On the left is a sidebar with navigation options: 'mapping', 'punti', 'reti', 'users', and a 'Crea tabella' button. The main area displays a table named 'reti'. Above the table, a green status bar indicates 'Mostrando i righi 0 - 3 (~4 totale) , La query ha impiegato 0.0006 sec'. Below this, the SQL query is shown: 'SELECT * FROM `reti` LIMIT 0, 30'. The table has columns: 'pid', 'idrete', 'ssid', 'mac', 'created_at', and 'updated_at'. There are four rows of data, each with a checkbox and action icons (Modifica, Modifica in linea, Copia, Elimina). Below the table, there are controls for 'Mostra' (30 righe a partire da # 0) and 'Modifica' / 'Elimina' / 'Esporta' buttons.

| | pid | idrete | ssid | mac | created_at | updated_at |
|--------------------------|-----|-------------|-----------|-------------------|---------------------|---------------------|
| <input type="checkbox"/> | 1 | 192.168.0.1 | Alice-102 | 06-25-44-2A-4B-F3 | 2013-05-30 16:19:45 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 2 | 192.168.0.1 | Vodafone | 00-07-64-3F-1D-F9 | 2013-05-30 16:28:37 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 3 | 192.168.0.1 | AlmaWi-Fi | 12-15-33-6C-2A-D3 | 2013-05-30 16:33:12 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 4 | 192.168.0.1 | NetGear | 61-11-34-2A-2A-A9 | 2013-06-05 13:43:24 | 0000-00-00 00:00:00 |

Tabella Reti

6.5 CREAZIONE/AGGIORNAMENTO TABELLA PUNTI

La tabella *Punti* si occupa di salvare tutti i punti localizzati dal sistema GPS memorizzando le coordinate, individuare la potenza del segnale wifi in un determinato punto ed assegnando ad ognuno di essi un id univoco.

| ID | LAT | LNG | RSSI | ID_RETE |
|----|-----------|-----------|------|---------|
| 1 | 41.689771 | 12.777615 | -33 | 2 |
| 2 | 39.612741 | 12.777615 | -51 | 3 |
| 3 | 41.68131 | 12.774954 | -38 | 2 |
| 4 | 44.491852 | 11.346817 | -60 | 1 |
| 5 | 42.433812 | 11.231445 | -44 | 4 |

```
CREATE TABLE punti (  
    pid int(11) primary key auto_increment,  
    lat varchar(100) not null,  
    lng varchar(100) not null,  
    rssid varchar(100),  
    idrete varchar(100),  
    created_at timestamp default now(),  
    updated_at timestamp  
);
```

Creazione di un Punto

Nel lato server, per creare un nuovo punto si utilizza il file php chiamato *crea_punto.php*. Questo file viene usato principalmente per la creazione di un nuovo punto

nelle relativa tabella.

Ciò avviene prendendo l'ID, SSID e il MAC di un punto, tali parametri vengono passati al database online tramite il metodo POST attendendo come risposta il tag successo/fallimento.

Nel codice si esegue una lettura dei dati dei punti tramite il metodo POST e la loro memorizzazione nella propria tabella. Alla fine si esegue una risposta adeguata *JSON*.

La risposta sarà: parametro mancante, punto aggiunto alla tabella o errore di inserimento dati.

Aggiornamento di un Punto

Si è creato un file php chiamato **update_point.php**. Ogni prodotto è identificato da PID. Per aggiornare un punto si effettua un metodo POST tramite PHP all'url dei punti presenti nella Tabella Punti.

Si attende un tag successo/fallimento. Se successo, permette di modificare i punti presenti nel database.

La risposta *JSON* sarà: successo file aggiornato o fallimento possibile campo mancante.

The screenshot shows the phpMyAdmin interface for the 'mapping' database, specifically the 'punti' table. The table is displayed with 6 rows and 7 columns: pid, lat, lng, rssid, idrete, created_at, and updated_at. The interface includes a sidebar with navigation options and a main area with a SQL query editor and table view controls.

| | pid | lat | lng | rssid | idrete | created_at | updated_at |
|--------------------------|-----|-----------|-----------|-------|--------|---------------------|---------------------|
| <input type="checkbox"/> | 1 | 41.689771 | 12.777615 | -33 | 2 | 2013-06-01 18:22:08 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 2 | 39.612741 | 12.777615 | -51 | 3 | 2013-06-03 13:30:00 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 3 | 41.68131 | 12.774954 | -38 | 2 | 2013-06-05 13:44:02 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 4 | 44.491852 | 11.346817 | -60 | 1 | 2013-06-10 15:17:58 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 5 | 42.433812 | 11.231445 | -44 | 4 | 2013-06-10 15:17:58 | 0000-00-00 00:00:00 |
| <input type="checkbox"/> | 6 | 42.134412 | 11.31545 | -54 | 4 | 2013-06-10 15:17:58 | 0000-00-00 00:00:00 |

Tabella Punti

6.6 CREAZIONE/AGGIORNAMENTO TABELLA USERS

La tabella *Users* ha lo scopo di avere una lista di utenti da poter gestire e stampare la posizione sulla mappa, a partire dalle coordinate. Contiene un id univoco, unique id (), un nome, una e-mail e una password (criptata col metodo base64_encode), salt per decriptare la password. Questa tabella serve a realizzare un login per l'utilizzo di questo database per ciascun utente.

| ID | UNIQUE ID | NOME | E-MAIL | PASSWORD | SALT |
|----|---------------|---------|-------------------|------------------|------------|
| 1 | 4f074eca601fb | Alberto | alberto@gmail.com | RlMwysdf4dasfa.. | 6f71b4fb34 |
| 2 | 8801wi59kuc24 | Manuela | manu@gmail.com | CrOgTr+cf3erq+3. | 1f486a8ae1 |

```

CREATE TABLE users(
    uid int(11) primary key auto_increment,
    unique_id varchar(23) not null unique,
    name varchar(50) not null,
    email varchar(100) not null unique,
    encrypted_password varchar(80) not null,
    salt varchar(10) not null,
    created_at datetime,
    updated_at datetime null
);

```

The screenshot shows the phpMyAdmin interface for a database named 'mapping' on localhost:3306. The 'users' table is selected. The SQL query shown is: `SELECT * FROM 'users' LIMIT 0, 30`. The table structure and data are displayed below.

| | uid | unique_id | name | email | encrypted_password |
|--|-----|-------------------------|------------------|---------------------|---------------------|
| <input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina | 1 | 51af23bb0ca8e0.28799928 | alberto scarlino | alberto@gmail.it | RIMwy4KEmPDb/dHcus |
| <input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina | 2 | 51af4842663bf2.22980581 | manuela corina | manu@gmail.it | CrOgTi+YcP+7wPwQ7F |
| <input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina | 3 | 51af521b71d472.89850937 | mario rossi | rossi@g.it | 4urkGgVa53oWnToTXh |
| <input type="checkbox"/> Modifica <input type="checkbox"/> Modifica in linea <input type="checkbox"/> Copia <input type="checkbox"/> Elimina | 9 | 51b1cd33a28614.13673082 | alberto scarlino | albertoooo@gmail.it | fau4i2FAFFSq4eg2JEs |

Tabella Users

Non avendo a disposizione un server, ci siamo serviti di un “server locale su una macchina linux”. Si è reso opportuno aggiungere una classe PHP per la connessione al database

MySQL con lo scopo principale di aprire una connessione al database online e chiuderla ogni volta non sia necessaria. Quindi sono stati creati due file:

db_config.php – contenente le variabili di connessione al database

```
<?php
/* Tutte le variabili per la connessione al database */
define('DB_USER', "root"); // db user
define('DB_PASSWORD', ""); // db password (mention your db password here)
define('DB_DATABASE', "mapping"); // database name
define('DB_SERVER', "localhost"); // db server
?>
```

db_connect.php - un file di classe per la connessione al database

```
<?php
* Un file di classe per la connessione al database */
class DB_CONNECT {
    // constructor
    function __construct() {
        // connecting to database
        $this->connect();
    }
    // destructor
    function __destruct() {
        // closing db connection
        $this->close();
    }
}
```

```

/* Funzione di connessione con il database */
function connect() {
    // importazioni variabili
    require_once __DIR__ . '/config.php';
    // connessione al database
        $con = mysql_connect(DB_SERVER, DB_USER, DB_PASSWORD) or
die(mysql_error());
    // Selezione database
    $db = mysql_select_db(DB_DATABASE) or die(mysql_error()) or die(mysql_error());
    // ritorna connessione cursor
    return $con;
}
/* Funzione per la chiusura della connessione al database */
function close() {
    mysql_close();
}
}
?>

```

Le operazioni eseguite sul database sono: Creare, Leggere, Aggiornare e Cancellare Reti o Punti sul database online utilizzando PHP.

| Tabella | Azione | Righe | Tipo | Collation | Dimensione | Overhead |
|--------------------------------|---|-----------|---------------|--------------------------|-----------------|------------|
| <input type="checkbox"/> punti | Mostra Struttura Cerca Inserisci Svuota Elimina | 4 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| <input type="checkbox"/> reti | Mostra Struttura Cerca Inserisci Svuota Elimina | 5 | InnoDB | latin1_swedish_ci | 16.0 KiB | - |
| <input type="checkbox"/> users | Mostra Struttura Cerca Inserisci Svuota Elimina | 4 | InnoDB | latin1_swedish_ci | 48.0 KiB | - |
| 3 tabelle | Totale | 13 | InnoDB | latin1_swedish_ci | 80.0 KiB | 0 B |

Schermata Principale del Database Online

6.7 CLASSE JSON PARSER

JSON (Java Script Object Notation), è una classe scritta in PHP il cui scopo è quello di migliorare l'interfaccia di programmazione verso Web Services per lo scambio di informazioni e dati client/server. È stata usata per ottenere JSON da URL.

Questa classe supporta due metodi di richiesta HTTP GET e POST che specificano il modo in cui PHP riceve dall'utente i dati.

Questi sono metodi del protocollo HTTP che specificano l'azione che il client richiede al server, il server esegue tale azione sulla risorsa identificata dall'URL.

Il metodo *GET* consiste nel richiedere informazioni ad un server, inviando pochi

parametri tramite URL attraverso la stringa di query. È più indicato, in quanto se i parametri sono troppi o troppo lunghi, passarli su linea di comando, oltre che essere poco pratico, potrebbe addirittura essere impossibile, in quanto il sistema operativo del server potrebbe avere una lunghezza massima per i comandi.

Il metodo POST è stato concepito per inviare al server molte informazioni, senza un limite sulla quantità di dati da trasmettere e sul tipo ed in modo non direttamente visibile per l'utente da URL. Viene eseguito dal Client solo in risposta ad una form.

Dunque, la differenza tra i due metodi consiste nel fatto che il metodo POST sostituisce quello GET qualora la richiesta effettuata tramite GET superi il limite massimo di caratteri consentito dal server e le informazioni da inviare non sono solo di tipo alfanumerico.

Viceversa, GET può essere utilizzato non solo per richiedere informazioni ma anche per sottoporre dati in formato testuale tramite form ad un processo oppure, come succede in questo caso di realizzazione dell'applicazione, riversarli in un Database.

7. SPERIMENTAZIONE

7.1 SMARTPHONE

Il termine Smartphone apparve per la prima volta nel 1997 quando la Ericsson descrisse il suo GS88 Penelope uno Smart Phone.

Con questo termine s'intende quindi un cellulare che abbia anche le funzioni e le potenzialità di un computer in grado di operare con un vero e proprio sistema operativo come iOS, Android, Windows Phone, Symbian, Blackberry 10, Bada, Open webOS ecc.

Il primo Smartphone fu progettato dalla IBM nel 1992 chiamato Simon e commercializzato dalla Bellsouth a partire dal 1993. Simon era dotato , oltre alle comuni funzioni di telefono, da una rubrica, calendario, orologio, block notes , funzioni di e-mail e giochi. Inoltre si poteva scrivere sullo schermo utilizzando un pennino in dotazione, consultare allegati e-mail, navigare in internet con un browser per mobile.

Negli anni successivi , nei primi anni 2000, inizia la nascita e l'evoluzione degli smartphone.

Nel giugno del 2007 arriva sul mercato internazionale quello che sarà il più grande nemico del nascente robottino verde, il primo iPhone di Apple, che ha rivoluzionato il modo di concepire gli smartphone. In molti si aspettavano che Google rispondesse con un proprio

smartphone, ma la sorpresa in arrivo era molto superiore alle aspettative. Infatti rimasero tutti stupiti quando Google presentò un intero ecosistema, un sistema operativo capace di funzionare su molti dispositivi diversi tra loro.

Lo smartphone ha il requisito di garantire una connettività a Internet (nei protocolli crescenti di velocità Gsm 2G, Gprs, Edge, Umts 3G e ora Hsdpa 3,5G) e avere funzioni professionali come email ed editor di testo. Inoltre è touchscreen (schermo a contatto) e/o una tastiera qwerty (estesa come quella di un computer). Se un telefono permette di eseguire solo applet java non è ritenuto uno smartphone.

Gli smartphone offrono la possibilità di installare in memoria ulteriori applicazioni che aggiungono sempre nuove funzionalità (le famose app). Molte cose si possono quindi fare, da scattare foto a giocare, da usare il navigatore satellitare Gps a scaricare email e navigare in Internet senza limitazioni come ad esempio, con il multitouch bastano due dita per zoommare e ruotare il dispositivo in grado di cambiare visualizzazione in orizzontale e verticale grazie ai sensori e tanto altro.

7.1.1 IL PRIMO SMARTPHONE ANDROID

Il primo smartphone dotato di un sistema operativo android (v. 1.0) è stato il T-Mobile G1 prodotto dall'HTC e presentato il 23 Ottobre 2003 a New York. Il dispositivo era dotato delle principali caratteristiche: tastiera Qwerty, schermo touchscreen da 3.2 pollici con risoluzione di 320x480 pixel, supporto per la connettività 3G UMTS/HSDPA a 7,2 Mbit/s, 192 MB di ram e 256 MB di memoria flash.

Con questo primo smartphone inizia l'evoluzione di Android. In seguito è stato commercializzato HTC Magic, con versione Android 1.5 targata Google, e con caratteristiche simili al G1 ma con tastiera a livello Hardware. Da qui in poi inizia l'evoluzione di Android .

Secondo Wikimedia Foundation, il sistema operativo Android nel 2010 è riuscito a superare Symbian, l'incontrastato sistema operativo di Nokia per oltre 10 anni, vendendo nel mondo ben 32,9 milioni di smartphone contro i 30,6 milioni di Symbian. Dal 2008 Android è cresciuto anno su anno.

7.2 CARATTERISTICHE DISPOSITIVI USATI

Affinchè il progetto in questione possa funzionare, si richiede che il dispositivo su cui si sta lavorando abbia determinate caratteristiche. Ad esempio, deve essere dotato di un display-touch medio/grande per la visualizzazione della mappa, deve avere una connessione ad Internet.

Di seguito abbiamo una descrizione dettagliata del dispositivo su cui è stato possibile effettuare il lavoro svolto: HTC Wildfire

Le specifiche, secondo le notizie del sito ufficiale:

- Dimensioni 106.75 x 60.4 x 12.9 mm
- Risoluzione schermo: QVGA 240 x 320
- Grandezza schermo: 3,2 pollici
- Input: Touchscreen capacitivo multi-touch
- Fotocamera da 5.0 Megapixel con autofocus e flash LED
- GPS (Antenna interna)
- Bussola digitale
- CPU Qualcomm MSM7225, 528 MHz
- RAM: 384 MB
- ROM: 512 MB
- compatibile con microSD (SDHC)
- Sistema operativo: Android 2.1 e HTC Sense
- Wi-Fi (802.11b/g)
- Bluetooth 2.0 + EDR & A2DP

- micro-USB
- 3.5 mm audio jack, microfono, altoparlante
- Accelerometro
- Radio FM con RDS



HTC Wildfire

7.3 RISULTATI OTTENUTI

In questo paragrafo si è provveduto a distinguere i risultati ottenuti effettuando dei miglioramenti su un'applicazione esistente Wi-Fi Mapping.

Tra i vari punti in cui ci si poteva lavorare, si è ottenuto:

1. La porting per versione di Android precedenti alla 4.0 avvenuta con una semplice riga di codice in androidManifest.xml

```
<uses-sdk android:minSdkVersion="8"  
    android:targetSdkVersion="17"/>
```

2. Possibilità di utilizzare delle mappe in modalità offline.

Questo punto è tuttora discutibile in quanto si stanno avendo dei servizi che aprono le porte anche a tale soluzione, già spiegata precedentemente nel capitolo 3.3.

3. Sviluppo di un database online da affiancare a quello locale: in tal modo gli utenti potrebbero condividere le proprie registrazioni (solo quelle che non richiedono una password e quindi utilizzabili da tutti).

Questi sono stati i punti in cui si è riusciti ad ottenere dei risultati, i restanti non sono stati implementati.

7.4 TEST SU HTC WILDFIRE

Come spiegato nel Capitolo 6, abbiamo provveduto a implementare un Database direttamente sul Web. Qui, dimostriamo figurativamente ciò che si è ottenuto.

Oltre a dimostrare la creazione di tabelle strutturate di dati di ogni genere con campi di tipo testo, numeri e altro, viene creata un'interfaccia grafica usabile per la gestione di esse presentandosi come vere e proprie applicazione web.

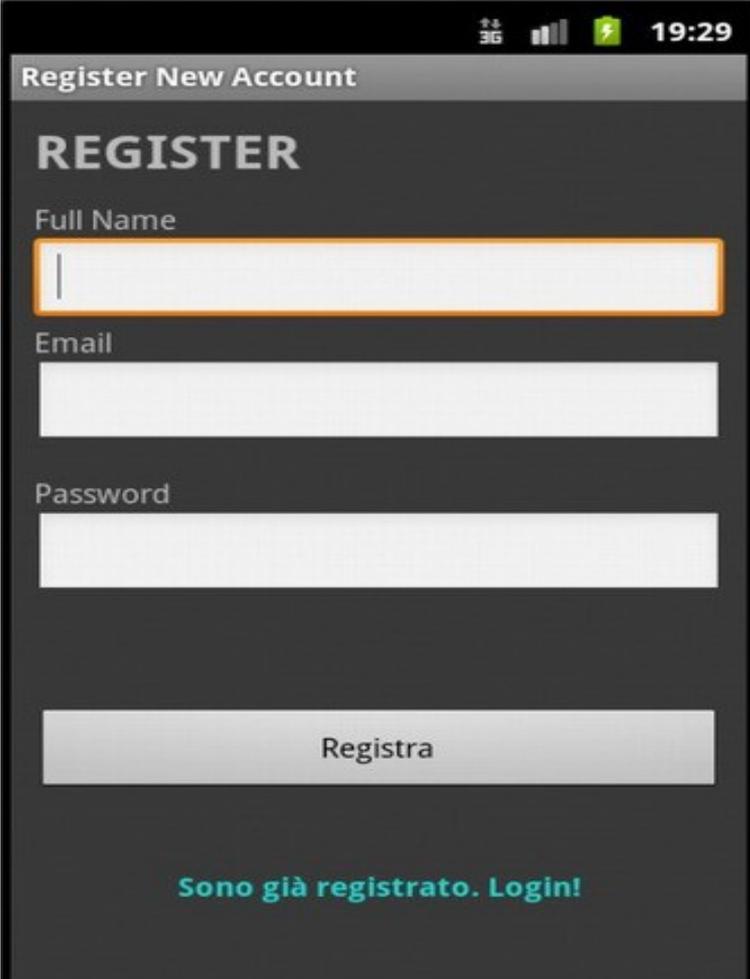
La condivisione svolge un ruolo importante proprio perché i dati sono visibili su Internet e quindi da altre persone che, con le dovute autorizzazioni, possono lavorare sui dati.

La protezione da accessi non autorizzati è un tema importante nella gestione dei database, che si rivelano vulnerabili nei seguenti punti:

- Sicurezza dei server.

- Connessioni tra database.
- Controllo sugli accessi al database.
- Data Warehouse (strumenti per localizzare, trasformare e caricare dati).
- Data Mining (insieme di tecniche per rendere utilizzabile e disponibile una struttura dati).

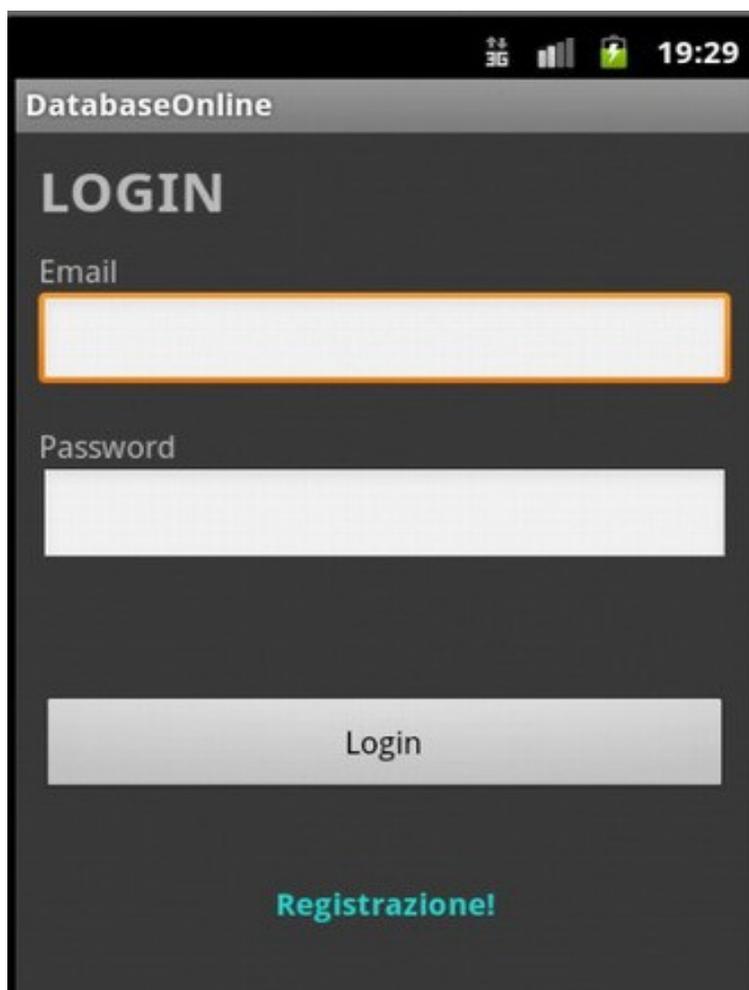
Per questo, per poter accedere al database online, è necessaria una Registrazione utente. Questa, come si può ben vedere, richiede l'immissione di un nome, una E-mail e Password.



The image shows a mobile application interface for registering a new account. At the top, the status bar displays '3G', signal strength, battery, and the time '19:29'. Below the status bar, the title 'Register New Account' is shown in a grey header. The main heading is 'REGISTER' in large, bold, white letters. There are three input fields: 'Full Name' (with a cursor), 'Email', and 'Password'. A 'Registra' button is located below the fields. At the bottom, there is a link that says 'Sono già registrato. Login!' in green text.

Registrazione Utente

Qualora un utente avesse già effettuato la registrazione, basta semplicemente effettuare un login.

The image shows a mobile application interface for a login screen. At the top, there is a status bar with icons for signal strength, battery, and the time 19:29. Below the status bar is a header with the text 'DatabaseOnline'. The main title of the screen is 'LOGIN' in large, bold, white letters. Below the title, there are two input fields: one labeled 'Email' and one labeled 'Password'. The 'Email' field is highlighted with a yellow border. Below the input fields is a large, light gray button labeled 'Login'. At the bottom of the screen, there is a link labeled 'Registrazione!' in red text.

Schermata Login

Ricordiamo che, nel momento in cui andiamo a creare la Tabella Utenti, per ciascuno di essi, abbiamo associato una Password criptata in SHA1.

SHA1 è un algoritmo in grado di produrre un message digest (messaggio d'impronta) di lunghezza fissa partendo da un messaggio di lunghezza variabile. La sicurezza di tale algoritmo consiste nel fatto che non sia possibile risalire al messaggio originale, essendo a conoscenza solo di questo parametro.

Fa parte della famiglia SHA-0, SHA-1, fino a SHA-3 usato in numerose applicazioni e protocolli, è molto efficiente in caso di attacchi esterni. Gli algoritmi SHA sono anche utilizzati negli algoritmi per la firma digitale dei documenti.

Successivamente, verrà aperto un Menù di Visualizzazione che ci permette di scegliere l'opzione desiderata.



Menù di Visualizzazione

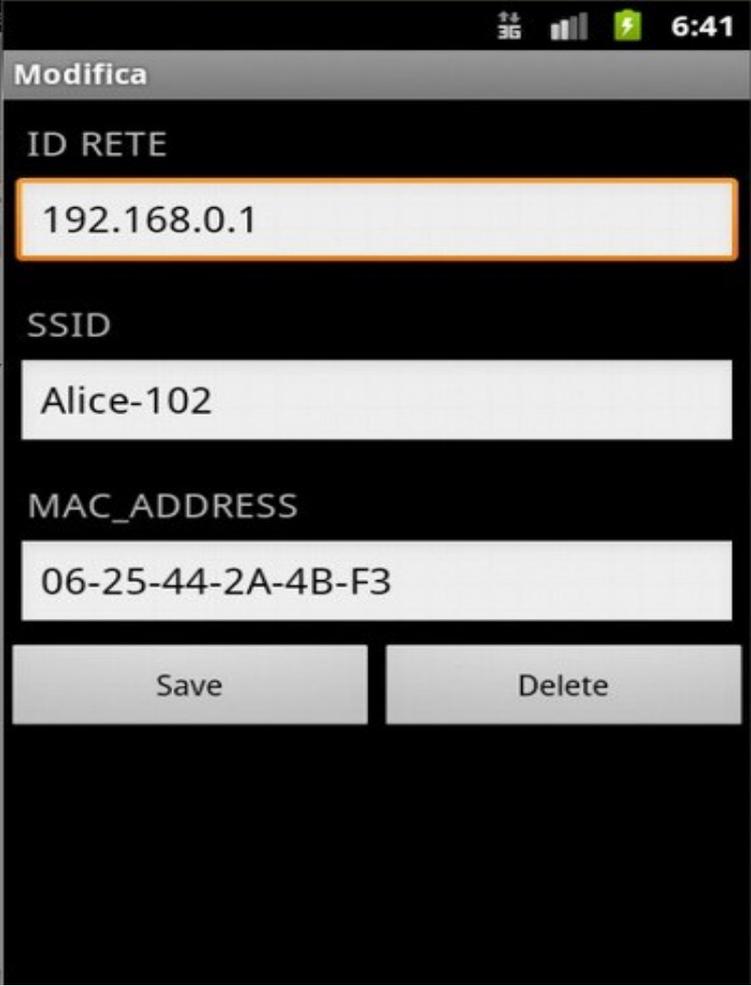
Se viene scelta l'opzione Aggiungi Reti, verrà aperta una finestra in cui ci vengono chiesti ID_RETE (ID della rete), SSID (nome della Wi-Fi che si identifica agli utenti) e MAC_ADDRESS (indirizzo fisico della Rete).

Creata la Rete, la si può visualizzare, scegliendo l'opzione Visualizza Rete, ottenendo la seguente schermata che ci permette di mostrare le reti presenti nel Database



Visualizza Rete

E' possibile effettuare delle modifiche o aggiornamenti riguardo la Rete.



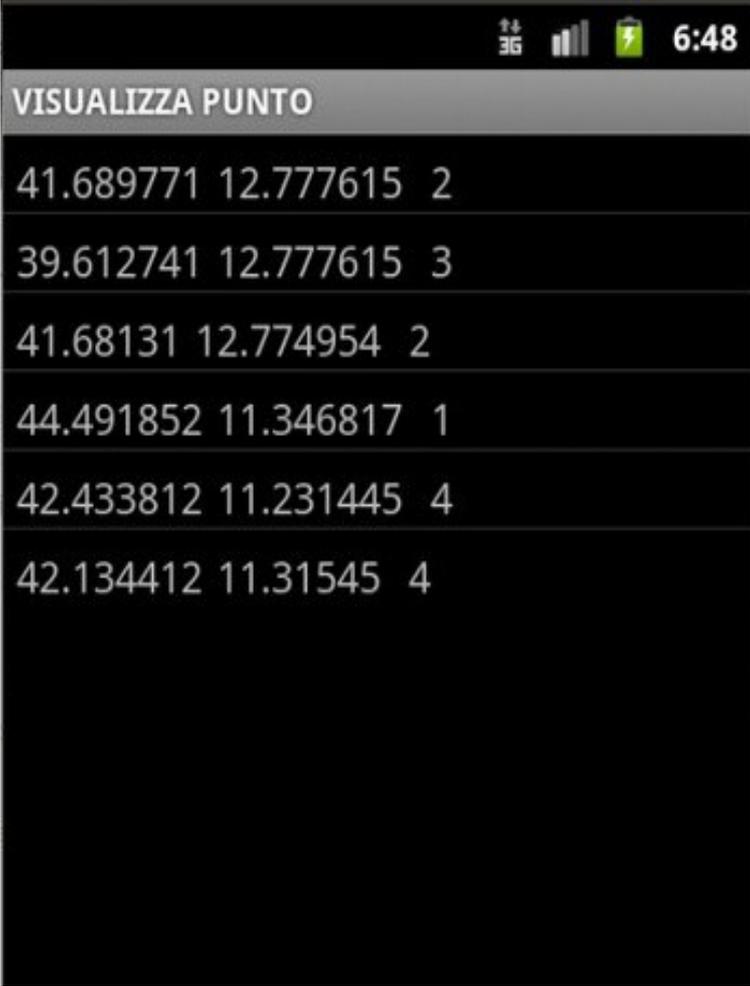
The screenshot shows a mobile application interface with a black background and white text. At the top, there is a status bar with icons for signal strength, battery, and the time 6:41. Below the status bar is a header labeled "Modifica". The main content area contains three text input fields: "ID RETE" with the value "192.168.0.1", "SSID" with the value "Alice-102", and "MAC_ADDRESS" with the value "06-25-44-2A-4B-F3". At the bottom, there are two buttons: "Save" and "Delete".

Modifica Rete

Dando, quindi, la possibilità oltre ad aggiornare i campi e salvare la rete, anche di eliminare la Rete. Ad esempio, nel momento in cui un Access Point è stato disattivato.

Se viene scelta l'opzione Aggiungi Punto, verrà aperta una finestra in cui ci vengono chiesti LAT, LNG, (che servono come coordinate GPS per trovare la posizione del dispositivo) RSSID, ID_RETE.

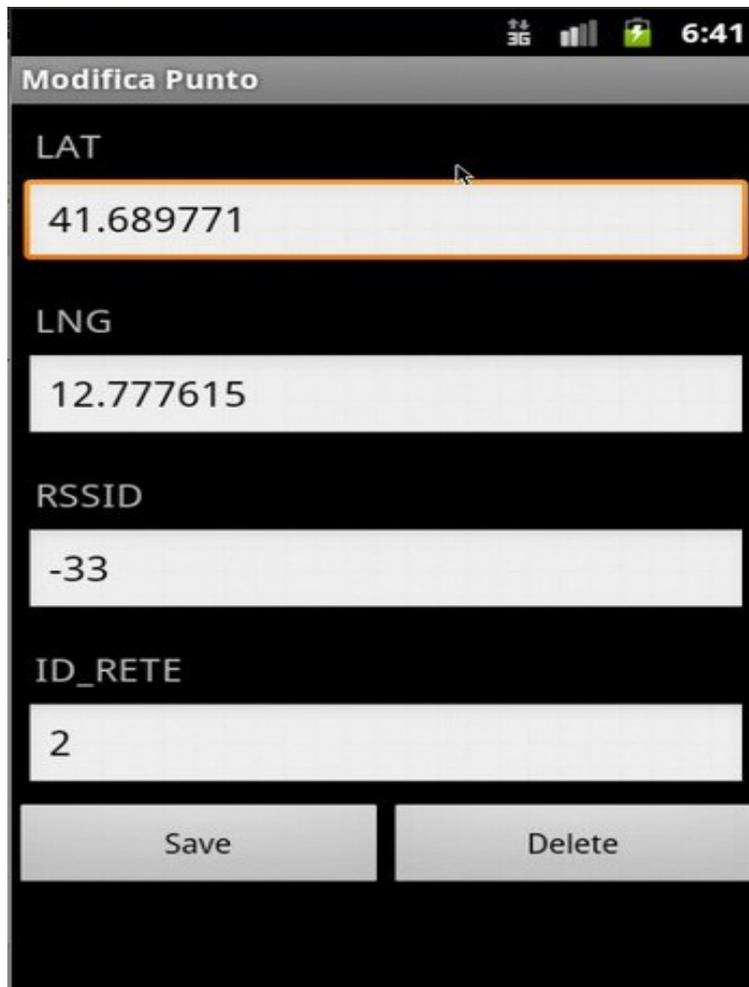
Creato il Punto, lo si può visualizzare, scegliendo l'opzione Visualizza Punto, ottenendo la seguente schermata



| VISUALIZZA PUNTO | | |
|------------------|-----------|---|
| 41.689771 | 12.777615 | 2 |
| 39.612741 | 12.777615 | 3 |
| 41.68131 | 12.774954 | 2 |
| 44.491852 | 11.346817 | 1 |
| 42.433812 | 11.231445 | 4 |
| 42.134412 | 11.31545 | 4 |

Visualizza Punto

Come per la rete, è possibile eseguire delle modifiche o aggiornamenti anche sui Punti



The screenshot shows a mobile application interface with a black background and white text. At the top, there is a status bar with icons for 3G, signal strength, battery, and the time 6:41. Below the status bar is a title bar labeled 'Modifica Punto'. The main content area contains four input fields, each with a label above it: 'LAT' with the value '41.689771', 'LNG' with '12.777615', 'RSSID' with '-33', and 'ID_RETE' with '2'. At the bottom of the screen, there are two buttons: 'Save' on the left and 'Delete' on the right.

Modifica Punto

CONCLUSIONI E SVILUPPI FUTURI

Giunti al termine di questa tesi, abbiamo notato che si è rivelata un interessante caso di studio su applicazioni e tecnologie all'avanguardia.

Lo studio di questo progetto ci ha fatto capire quanto, il mondo mobile, abbia aperto delle frontiere enormi per quanto riguarda l'elaborazione di informazioni di tutti i generi.

È stata fatta una panoramica generale di questi nuovi dispositivi soffermandoci in particolare su dispositivi Smartphone con sistema Android puntualizzando alcune specifiche come il kit di sviluppo di software Android SDK e file Android Manifest.. La semplicità di utilizzo e di sviluppo, offerta dalla piattaforma Android, ha permesso di lavorare in ambiti molto ampi e di alto livello di accesso e abilità, pur utilizzando risorse complesse come mappe e funzionalità di geolocalizzazione. Su questo, infatti, si sono mostrate nel dettaglio le procedure e gli strumenti che permettono di utilizzare gratuitamente gli svariati servizi offerti da Google Maps in particolare le mappe in modalità offline, che continua ad essere in grande evoluzione.

Ricordiamo che il lavoro, effettuato con la collega Manuela Corina, ha avuto una base di partenza fornitaci dai colleghi Cautela, Peca e Salpietro nell'implementare un'applicazione per la mappatura degli Access Point sui sistemi mobili.

Inizialmente, si è provveduto a scegliere con cura, le caratteristiche principali del software da utilizzare, scegliendo l'ambiente di sviluppo Eclipse con Android SDK e la piattaforma L.A.M.P. Tale Server si è reso molto importante per la creazione di un Database online in grado di creare o modificare Punti o Reti al fine di poter condividere registrazioni nel momento in cui ci si ha la possibilità di connettersi ad una rete Wi-Fi, attraverso cui si da

la possibilità a diversi terminali di potersi collegare tra loro. adatti a soddisfare i nostri requisiti.

Affinchè l'implementazione sia più facile, abbiamo fatto uso di API. Ovvero procedure già predisposte disponibili al programmatore con la finalità di ottenere un'astrazione a più alto livello, semplificando appunto il lavoro di dover riscrivere righe di codice più volte.

Il tutto è stato testato su un dispositivo di nostra proprietà come l'HTC wildfire.

In conclusione, il nostro lavoro ha permesso di migliorare solo in parte alcuni punti salienti di questa applicazione. Di seguito, si citano, quindi, ulteriori sviluppi che si possono effettuare in futuro:

- includere e affiancare il database online all'applicazione wifi mapping;
- migliorare la gestione delle reti, ovvero, possibilità di decidere se condividere, sul database online, reti private o pubbliche;
- effettuare un backup del database locale e online;
- ripristinare il backup;
- migliorare la gestione dell'aggiornamento reti.

BIBLIOGRAFIA

- [1] Google Maps offline: consultare le mappe da Android senza connessione dati:
www2.01net.it/articoli.asp?ID=9487
- [2] Google Maps ed Earth: www.tuttoandroid.net/android/google-maps-earth-mappe-hd-3d-59021/
- [3] Scheda Tecnica Wildfire http://it.wikipedia.org/wiki/HTC_Wildfire
- [4] Applicazioni iOS e Android con Google Maps, autore Giacomo Andreucci, Volume 1
- [5] Applicazioni iOS e Android con Google Maps, autore Giacomo Andreucci, Volume 2
- [6] Android Login and Registration with PHP, MySQL and SQLite:
www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/
- [7] Come connettersi ad Android con PHP-MySQL: www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/
- [8] Google Maps Android API v2 – Tutorial:
www.vogella.com/articles/AndroidGoogleMaps/article.html
- [9] Google Maps Android v2 API:
https://developers.google.com/maps/documentation/android/start#installing_the_google_maps_android_v2_api
- [10] Google Maps Android v1 API:
<https://developers.google.com/maps/documentation/android/v1/>
- [11] Algoritmo SHA1: wikipedia.org/wiki/Secure_Hash_Algorithm