Alma Mater Studiorum · Università di Bologna

facoltà di scienze matematiche, fisiche e naturali

Corso di Laurea Magistrale in Informatica

Applicazioni di Web 2.0 in ambiente iOS

Tesi per il corso di Sistemi Distribuiti

Relatore:
Prof.
Alessandro Amoroso

Presentata da:
Niccolò Dionisi

Sessione
Luglio 2011

Alla mia famiglia
e a chi crede in me

Introduzione

In questi ultimi anni gli *smartphone*, ovvero quei dispositivi che offrono oltre alle caratteristiche solite di un cellulare anche altre più complesse simili a quelle di un computer, stanno conquistando quote sempre maggiori del mercato della telefonia mobile, aumentando di anno in anno.
A questo bisogna aggiungere l'arrivo, in tempi ancora più recenti, di *tablet computer* che si vanno a collocare, almeno per alcuni utilizzi, nella fascia di mezzo tra gli smartphone ed i computer portatili fondendo le caratteristiche di entrambi in un unico apparecchio più completo di uno smartphone e più maneggevole ed intuitivo di un laptop.
Queste nuove tecnologie fanno sì che la ricerca di nuovi software sia sempre più rivolta verso il mondo dei dispositivi mobili.

Tutti questi dispositivi di ultima generazione hanno un sistema operativo che spesso deriva da quelli utilizzati per i computer portatili, ma privato di tutte le funzionalità superflue, per renderlo così veloce, leggero ed intuitivo. Inoltre offrono tutti una connessione ad internet sia utilizzando il Wi-Fi integrato che il 3G abbonandosi ad un piano tariffario di una compagnia telefonica e permettono di installarvi migliaia di applicazioni diverse, utili e meno utili, sviluppate dal produttore dello stesso smartphone o da terze parti, che possono essere scaricate gratuitamente o pagando una piccola cifra.

L'idea alla base di questo progetto di tesi è stata in primo luogo quella di creare un'applicazione per dispositivi mobili che portasse qualche elemento

nuovo e originale, non ancora sviluppato sul panorama attuale.
Tra le tante opportunità disponibili in questo campo ho scelto di realizzare la tesi in ambiente Apple sia per la grande distribuzione che dispositivi come iPhone e iPad stanno avendo all'interno del mercato sia come interesse personale poiché, non essendomi mai occupato di progetti che rientrassero in uno di questi ambiti, è stata per me una sfida dettata anche dalla curiosità.
Inoltre la Apple permette, registrandosi come sviluppatore, di vendere direttamente attraverso l'App Store le proprie applicazioni agli utenti senza intermediari. In questo modo è possibile che un'idea geniale riesca ad ottenere un buon successo anche senza investirvi ingenti quantitativi di denaro.

Molti smartphone poi sono dotati della tecnologia GPS che rende possibile la realizzazione di applicazioni che permettano all'utente, in pochi istanti, di vedere la propria posizione geografica visualizzata su di una mappa.
A questo va aggiunto che grazie agli schermi sempre più grandi e definiti, quasi totalmente multi-touch, l'utente ha la possibilità di scorrere le mappe, zoomare e spostarsi, con pochi gesti delle dita.
Poi, semplicemente selezionando un pulsante è in grado di condividere con gli altri la propria posizione.
La condivisione, appunto, e l'interazione tra gli utenti sono diventate fondamentali nell'era del Web 2.0 dove i contenuti presenti nella rete vengono continuamente creati, modificati, e discussi dagli utenti sparsi in giro per il mondo, senza il bisogno che qualcuno, dall'alto, li gestisca.
Proprio unendo l'interesse suscitato da queste nuove possibilità alle considerazioni precedenti, si è deciso di realizzare un'applicazione per iPhone che consentisse l'individuazione di gruppi di persone nell'ambiente circostante, combinando quindi la geolocalizzazione dell'utente con l'aspetto sociale della condivisione di alcune informazioni.

In particolare l'applicazione non si occuperà di mostrare i movimenti di singoli individui, come altre già permettono di fare, ma di presentare a coloro

che la utilizzeranno i luoghi in cui vi è una maggior concentrazione di persone, siano essi spazi pubblici o locali privati.
Contrariamente a come funzionano, ad esempio, i servizi messi a disposizione da alcuni *Social Network* grazie ai quali l'utente aggiorna (volontariamente o in maniera automatica) la propria posizione e questa viene condivisa a tutti i suoi contatti, nel nostro caso ogni utente contribuirà, fornendo la propria posizione, ad aggiornare dei luoghi "globali".
I dati forniti dall'utente saranno utilizzati assieme a quelli di tutti gli altri per poter restituire informazioni dettagliate ma di carattere generale.

Una delle caratteristiche fondamentali che distinguono questa applicazione sarà infatti quella dell'anonimato. Questo significa che, diversamente da altre applicazioni simili già esistenti e molto diffuse sul mercato che richiedono l'iscrizione e l'autenticazione ad un servizio rendendo in questo modo disponibili i propri dati personali, in questo progetto non è presente alcun processo di registrazione o autenticazione.
Ogni utente che utilizzi l'applicazione invierà i dati relativi alla propria posizione in quel momento per poter vedere quelle di altri utenti ma senza sapere dove si trova la tal persona bensì semplicemente ricevendo in risposta dal server i luoghi nell'ambiente circostante in cui sono presenti altre persone senza conoscere l'identità di queste.
Un utente, senza violare la privacy altrui e senza distribuire in giro i propri dati, può così venire a conoscenza dei luoghi più (e meno) frequentati nella zona in cui si trova per poi organizzarsi di conseguenza.
Questa caratteristica, nonostante possa sembrare marginale, risulta invece essere fondamentale poiché mentre la maggiorparte delle più famose applicazioni "sociali", soprattutto quelle per dispositivi mobili, tendono a distribuire in giro i dati personali dell'utente rendendoli disponibili a tutti, qui si è voluto appositamente mettere un freno a ciò creando quindi un servizio libero senza bisogno di procedure di registrazione o di immissione di dati personali. Si riesce così anche a ridurre la possibilità di un uso malevolo dell'applicazio-

ne poiché i dati restituiti dal server non possono essere utilizzati in nessun modo per scopi diversi da quelli di informazione, non conoscendo l'identità delle persone che affollano i luoghi.
Il fulcro del sistema non risulterà quindi più essere la singola persona ma il gruppo, inteso come insieme di persone che condividono lo stesso luogo in un determinato momento.
L'obiettivo generale che si vuole raggiungere è quello di fare inviare all'utente i propri dati non per interesse personale ma per contribuire a migliorare uno strumento dedicato alla comunità, per sentirsi parte di un gruppo di persone ed aiutare gli altri utenti a meglio destreggiarsi negli spostamenti. Poiché questo spesso non accade, nell'applicazione qui descritta si è deciso di "costringere" l'utente ad inviare i propri dati in forma anonima per poter vedere i gruppi di persone attorno alla sua posizione geografica.

All'utente verrà solamente chiesto di fornire, facoltativamente, dei dati da utilizzare per elaborazioni statistiche come età e sesso, affinché questi possano essere poi usati nelle ricerche, magari assieme ad altri aggiunti in un secondo momento, per far scoprire a chi utilizza l'applicazione quali sono le location più adatte ai propri gusti e all'ambiente sociale cui appartiene.
Ogni utente potrà inoltre inviare messaggi relativi al luogo in cui si trova, sempre anonimamente, per aiutare chi è alla ricerca di un posto dove andare attraverso critiche e consigli o semplicemente per informare le altre persone che vogliono informazioni su quel luogo di che cosa sta accadendo in quel momento.

L'idea è sia quella di fornire all'utente uno strumento per potersi destreggiare in una città a lui sconosciuta sia quella di potere evitare posti troppo affollati.
Ad esempio una persona che si trova in vacanza in una grande città ed ha voglia di incontrare gente potrà utilizzare l'applicazione per cercare i luoghi dove c'è più vita in quel particolare momento della giornata. Non dovrà fa-

re altro che avviarla, inviare i propri dati e inserire come criterio di ricerca il massimo numero di persone. Nella mappa sul suo iPhone compariranno tante location con indicato il numero di persone ed egli potrà decidere di conseguenza.
Utilizzando la stessa applicazione, una persona che non vuole confusione, potrà osservare quali sono i luoghi più affollati per poterli così evitare.
E questi sono solo alcuni esempi di utilizzo che un'applicazione di questo tipo può fornire ai propri utenti.

Naturalmente in questi casi partiamo dal presupposto che un gran numero di persone posseggano un dispositivo Apple, abbiano installato l'applicazione e la utilizzino in contemporanea.
Come si può facilmente intuire, applicazioni di questo tipo non hanno nessun dato già caricato al momento dell'avvio ma utilizzano dati creati, gestiti e modificati direttamente o indirettamente dagli utenti. In particolare in quest'applicazione il dato principale sono gli utenti stessi che inviando la loro posizione incrementano o decrementano il numero di utenti nelle varie location. Si ha quindi bisogno di un sempre maggior numero di utenti attivi per poter garantire lo scopo per cui sono state create.
Spesso infatti vengono lanciate a livello locale, rendendo disponibili i servizi che forniscono per una sola zona o città. Se col passar del tempo riescono poi a raggiungere notorietà ed un numero sempre maggiore di utenti si allargano fino ad arrivare, in certi casi, a coprire l'intero globo.
Ma, come è giusto ricordare, lo scopo di questa tesi è stato quello di studiare, progettare ed implementare un'alternativa nuova e originale alle moderne applicazioni per dispositivi mobili senza curarsi, almeno in questo momento, dei problemi che sorgono nelle fasi di distribuzione e commercializzazione.

Si è riusciti nell'intento di creare un'applicazione che combinasse tutte le caratteristiche precedentemente descritte grazie alle tecnologie messe a disposizione dall'ambiente di sviluppo Xcode, creato dalla stessa Apple, insie-

me a framework esterni contenenti funzione specifiche per la gestione delle mappe e delle coordinate dell'utente. Questo per quanto riguarda la parte di software installata sul dispositivo.
Essa è stata fatta lavorare in congiunzione con un server su cui è stato installato un database capace di gestire coordinate geografiche e delle pagine web dinamiche per permettere al dispositivo di comunicare con la base di dati per inviare e ricevere i dati degli utenti.

Arrivando infine a questo documento di tesi, utile per comprendere meglio tutti i diversi passaggi che si sono seguiti nei mesi di lavoro, nelle pagine che seguono viene descritto inizialmente lo stato dell'arte, presentate le metodologie generali e il quadro di riferimento in cui si va ad inserire questa applicazione.
Per far ciò verrà esposto lo stato attuale dei dispositivi mobili ed in particolare degli smartphone, dei loro sistemi operativi e della distribuzione digitale di software con un occhio particolare ad Apple, descrivendo poi alcune applicazioni già esistenti sul mercato che hanno caratteristiche simili o in comune a quella descritta in questo documento, per far meglio capire quali sono gli aspetti più apprezzati dagli utenti e dal mercato.
Sarà così presentato il progetto delineandone i requisiti, la struttura, le metodologie adottate per portarlo a compimento ed il suo funzionamento prima in generale, successivamente nel dettaglio.
Qui vengono elencate anche alcune problematiche o dubbi progettuali che sono stati poi risolti prima di poter implementare al meglio il software.
Nella parte relativa ai principali aspetti implementativi vengono mostrati gli strumenti utilizzati e motivato brevemente il perché della loro scelta per poi entrare nel dettaglio dell'implementazione facendo anche osservare, quando necessario, parti di codice relative alla parte prettamente applicativa e alla parte server. Inoltre verrà presentata una sezione di sperimentazione contenente le prove ed i test effettuati per studiare l'implementazione del progetto e capire se apportare ulteriori modifiche.

Al termine di questa sezione troviamo l'elenco delle principali problematiche implementative riscontrate durante la fase di testing e debugging e gli approcci alla loro risoluzione.
Il documento termina con un resoconto del lavoro svolto per arrivare alle conclusioni che sono state tratte oltre ai possibili sviluppi futuri dell'applicazione o di parti di essa.

Indice

Elenco delle figure

Capitolo 1

STATO DELL'ARTE

In questo capitolo vengono presentate le metodologie generali e il quadro di riferimento in cui si colloca l'applicazione.
Per poter tracciare una panoramica generale del mondo in cui questa applicazione va a collocarsi, si è deciso di iniziare con una descrizione dei dispositivi attualmente in circolazione nonché dei sistemi operativi in essi contenuti che permettono al software di essere utilizzato.
Una sezione è stata dedicata anche alla distribuzione digitale di software poiché sta riscontrando un successo sempre maggiore nella vendita delle applicazioni in particolare per dispositivi mobili ma anche, negli ultimi tempi, per laptop e computer fissi.
Si passa infine alla descrizione di alcune applicazioni già esistenti sul mercato che, per alcune loro caratteristiche, appartengono alle tipologie che forniscono servizi simili a quella qui descritta ed aiutano a comprendere meglio quali sono le scelte degli utenti al momento di decidere di utilizzare un'applicazione rispetto ad un'altra.

1.1 Smartphone

Uno *smartphone* è un dispositivo mobile che offre abilità di computazione e connettività molto maggiori se confrontato con un telefono cellulare di

ultima generazione.
Infatti, nonostante i telefoni cellulari più moderni riescano ad utilizzare applicazioni basate su una piattaforma come può essere *Java ME*, uno smartphone si avvale di un sistema operativo completo che fornisce una piattaforma per sviluppatori di applicazioni di terze-parti.
In questo modo gli utenti di uno smartphone possono utilizzare, anche contemporaneamente, più applicazioni create appositamente per l'hardware di quel dispositivo.
Uno smartphone combina quindi assieme le funzioni di un PDA (*personal digital assistant*) ed un telefono cellulare, che solitamente è un telefono portatile provvisto di fotocamera e GPS.
Per quello che riguarda la connettività questi apparecchi utilizzano connessioni di tipo GSM/GPRS/EDGE/UMTS/HSDPA/HSUPA per la telefonia e l'accesso ad internet oltre a tecnologie come Bluetooth e Wi-Fi per le comunicazioni con altri dispositivi.

La crescita della domanda per dispositivi mobili avanzati ha fatto sì che essi montino processori sempre più potenti oltre alla memoria (Flash) sempre più abbondante ed a schermi ad alta risoluzione con capacità multi-touch per permetterne l'utilizzo anche in ambito ludico.

Secondo diversi rapporti, negli ultimi anni gli smartphone stanno sperimentando tassi di adozione sempre maggiori.
Ad esempio la *Berg Insight* ha riportato nel marzo 2011 il dato che mostra come la spedizione globale di smartphone tra il 2009 e il 2010 è incrementata del 74%. [3]

Come già detto è un sistema operativo mobile, o mobile OS, a controllare i dispositivi mobili, in linea di principio come un sistema operativo tipo Windows, Mac OS, o Linux controlla un computer desktop o portatile.
I sistemi operativi per smartphone però sono resi molto più semplici e scarni

sia perché devono lavorare su dispositivi con caratteristiche più limitate sia perché, essendo utilizzati spesso mentre si è fuori casa e ci si sposta da un luogo ad un altro, gli utenti richiedono velocità di computazione e facilità di utilizzo.
La crescente importanza dei dispositivi mobili ha messo in moto una forte competizione tra alcuni giganti del software come Google, Microsoft e Apple, insieme alle industrie leader nel settore mobile come Nokia, Research In Motion (RIM) e Palm per cercare di catturare subito la fetta di mercato più ampia. Sono nati così diversi sistemi operativi che si evolvono di pari passo con i dispositivi su cui vengono installati. I più noti che si contendono il mercato sono Symbian, Android, iOS, BlackBerry OS, Windows Phone, Linux, webOS, Bada, Maemo e MeeGo tra gli altri. Tutti questi sistemi operativi hanno peculiarità che li fanno differire dai concorrenti ma allo stesso tempo cercano di inglobare le caratteristiche più riuscite degli altri rinnovandosi in tempi sempre più brevi.

Poiché l'applicazione è stata sviluppata per il sistema operativo iOS della Apple ed in particolare per il dispositivo iPhone, qui di seguito verrà focalizzata l'attenzione su di essi.

1.1.1 iPhone e iOs

L'iPhone, insieme ad iPod ed iPad utilizza iOS (precedentemente conosciuto come iPhone OS), una versione ottimizzata del sistema operativo Mac OS X a cui sono state tolte alcune componenti non necessarie. Entrambi hanno il nucleo basato sul sistema *Darwin*, a sua volta derivato da una fusione tra FreeBSD e Mach. Risultano quindi essere sistemi della famiglia *UNIX-like* ma, a differenza di Android, solo il nucleo può essere definito *Open Source*.
Il processore dell'iPhone appartiene alla famiglia ARM ed il sistema operativo, a differenza di quello per computer che è compilato per processori PowerPC e X86, è compilato per questa tipologia di processori. Di conseguenza non è possibile copiare semplicemente le applicazioni Mac OS X da un

PC ad un dispositivo mobile, ma queste devono essere riscritte e ricompilate per poter funzionare correttamente. Il sistema operativo occupa all'incirca 700 MB di spazio nella memoria flash del dispositivo.
Dopo la commercializzazione iniziale del prodotto sono stati riscontrati alcuni problemi di stabilità e sicurezza, che sono stati risolti con il primo aggiornamento rilasciato il primo agosto 2007 dalla Apple. Quello stesso aggiornamento ha anche eliminato qualsiasi modifica o hack del telefono.
L'11 luglio 2008 è stata rilasciata la versione 2.0 del sistema operativo ed in contemporanea l'iPhone 3G è stato reso disponibile in 22 paesi.
È stato così introdotto l'App Store, un negozio on-line tramite il quale gli utenti possono scaricare gratuitamente o a pagamento applicazioni per il dispositivo.
Il 7 giugno 2010, in contemporanea alla presentazione del nuovo iPhone 4, è stata annunciata la quarta versione del sistema operativo, iOS 4.0, pubblicata ufficialmente il 21 giugno.
Il 23 novembre 2010 è stata rilasciata la versione 4.2.1 che per la prima volta ha portato tutte le funzioni di iOS 4 anche su iPad. [2]

Dal 2007, anno di nascita del primo iPhone ad oggi si sono seguite diverse generazioni di questo telefono. L'ultima, a questo momento, è l'iPhone 4. Le principali differenze tra questo ed i suoi predecessori, oltre al design, sono le componenti interne. Esso monta un processore Apple A4 e 512 MB di eDRAM, il doppio dei quella del suo predecessore e il quadruplo di quella dell'iPhone originale. Il display da 3.5 pollici a LED ha una risoluzione di 960 x 640 pixel ed è commercializzato come *Retina Display*.
Per stare dietro alla concorrenza e alle esigenze di mercato questi dispositivi ed i relativi sistemi operativi vengono aggiornati con una frequenza sempre maggiore costringendo anche gli sviluppatori di applicazioni a continui aggiornamenti per non perdere parte dell'utenza.

Per quello che riguarda l'applicazione descritta in questo documento, una

volta optato per l'ambiente Apple, si è scelto di svilupparla per il dispositivo iPhone poiché l'iPod touch, nonostante condivida le stesse caratteristiche e lo stesso sistema operativo, manca del GPS e quindi non può essere utilizzato per la geolocalizzazione se non attraverso il collegamento Wi-Fi ad internet che però non è sempre disponibile ed è meno preciso di quello GPS.
Relativamente all'iPad, l'applicazione può essere facilmente trasferita su questo dispositivo con le opportune modifiche ai sorgenti del programma (già predisposti) ma, essendo un'applicazione pensata per essere utilizzata dall'utente mentre questo si sposta per la città, non risulta tanto comodo l'utilizzo su un dispositivo della grandezza dell'iPad.

Ricapitolando l'iPhone, rispetto agli altri device di casa Apple, ha tutte le caratteristiche necessarie al funzionamento del nostro programma oltre ad essere più piccolo e maneggevole. In particolare adotta le tecnologie UMTS e HSDPA che vengono utilizzate anche per la connessione ad internet oltre ad includere un dispositivo *Assisted GPS* che permette la localizzazione geografica dell'utente, fondamentale per la nostra applicazione.
Inoltre il display più grande di quello di altri smartphone e multi-touch permette una perfetta interazione con le mappe fornite dal *Map Kit Framework* la cui visualizzazione può essere facilmente modificata scorrendo o zoomando con il tocco delle dita.

1.2 Distribuzione digitale

La distribuzione digitale è la pratica di consegnare contenuti senza l'utilizzo di mezzi fisici, solitamente scaricandoli da internet direttamente sul dispositivo dell'utente. Questo metodo di distribuzione, sempre più utilizzato, bypassa i metodi di distribuzione convenzionali.
In particolare nel nostro caso si tratta di download di software (applicazioni) per dispositivi mobili. Le principali piattaforme di distribuzione digitale che operano in questo campo sono l'App Store della Apple per i dispositivi con

il sistema operativo iOS, l'Android Market di Google per quelli che hanno Android, Ovi Store per i dispositivi nokia e Windows Phone Marketplace per quelli con Windows Phone 7.
Ognuno di questi servizi è accessibile attraverso un'applicazione preinstallata sul dispositivo e permette di scaricare gratuitamente o a pagamento migliaia di applicazioni disponibili, pubblicate da sviluppatori di terze-parti.
Il mercato digitale delle applicazioni mobili ha avuto negli ultimi anni un boom spaventoso e, secondo una recente analisi, entro il 2015 raggiungeranno il ragguardevole volume di 182,7 miliardi di download, in crescita dai 10,7 miliardi registrati nel corso del 2010. [4]
Questo mercato sta subendo una variazione soprattutto per quanto riguarda le applicazioni gratuite che permettono di fare acquisti così detti *in-app*, all'interno dell'applicazione stessa. L'utente scarica ed installa gratuitamente un'applicazione che però ha delle funzionalità limitate. Per sbloccarne altre o ottenere caratteristiche aggiuntive l'utente deve acquistarle.
In generale, con questo sistema di distribuzione, l'utente non avrà più bisogno di girare per il web alla ricerca di un software che soddisfi i propri bisogni ma sarà sufficiente accedere al negozio digitale, scorrere tra le categorie e scegliere l'applicazione più vicina alle sue esigenze e alle sue tasche.

In particolare qui di seguito verrà fatta una breve panoramica sull'attuale situazione dell'App Store di Apple, essendo la nostra applicazione sviluppata per iPhone.

1.2.1 App Store

L'App Store è un servizio realizzato da Apple disponibile per iPhone, iPod touch e iPad che permette agli utenti di scaricare e acquistare applicazioni disponibili in *iTunes Store*. Le applicazioni possono essere sia gratuite che a pagamento, e possono essere scaricate direttamente dal dispositivo o su un computer.
L'App Store è stato aperto il 10 luglio 2008 tramite un aggiornamento soft-

ware di iTunes.
Volendo fornire qualche dato su quello che è il mercato relativo a questo servizio, il 10 luglio 2008, il CEO Steve Jobs ha dichiarato che l'App Store conteneva 500 applicazioni di terze parti per iPhone e iPod touch, e il 25% di queste erano gratis. Le applicazioni vanno dall'intrattenimento all'istruzione. L'11 luglio 2008, l'App Store viene aperto permettendo agli utenti di acquistare le applicazioni tramite il proprio dispositivo. Nella prima settimana sono state scaricate 10 milioni di applicazioni.
Il 16 gennaio 2009, Apple ha annunciato sul proprio sito web di aver raggiunto il traguardo di 500 milioni di applicazioni scaricate. La miliardesima applicazione è stata scaricata il 23 aprile 2009.
A differenza delle applicazioni native presenti su iPhone, iPod touch e iPad, le applicazioni scaricate dall'App Store possono essere rimosse. [5]
In poche parole un utente può entrare nello store tramite l'applicazione predisposta sul suo cellulare. Qui può scorrere le applicazioni, vederne le caratteristiche e le opinioni degli altri utenti.
Utilizzando l'account Apple a cui è associata la carta di credito per il pagamento egli può acquistare l'applicazione che viene direttamente scaricata ed installata sul proprio iPhone e può essere subito utilizzata.
Il 20 ottobre del 2010 è stato annunciato il rilascio del *Mac App Store*, del tutto uguale all'App Store per dispositivi mobili, ma relativo ad applicazioni per il sistema operativo Mac OS X e quindi laptop e Mac fissi.
Così, anche chi non dispone di un dispositivo mobile, può utilizzare questo sistema per ottenere software per il proprio computer direttamente da internet.
Questo fa capire come, almeno secondo Apple, questo sistema rappresenti il futuro per la vendita e la distribuzione di qualsiasi tipo di prodotto che non abbia bisogno di un supporto fisico.
Infine al 22 gennaio 2011 si è calcolato che erano disponibili in App Store più di 325.000 applicazioni sviluppate da terze parti, con oltre 10 miliardi di download effettuati.

L'applicazione descritta in questo documento, essendo sperimentale e non ancora testata sul dispositivo non è stata inserita nello store della Apple. La Apple stessa inoltre impone che, per poter sia testare l'applicazione su dispositivo che renderla disponibile sull'App Store, l'autore della stessa abbia un account da sviluppatore che viene rilasciato sotto il pagamento di una quota annua.
È comunque molto interessante questo sistema di vendita e distribuzione delle applicazioni poiché permette a chiunque sia uno sviluppatore, anche senza una ditta alle spalle, di rendere in pochi passi la propria applicazione disponibile a migliaia di utenti in tutto il mondo senza intermediari. In questo modo un'idea innovativa, che altrimenti per mancanza di fondi o pubblicità sarebbe rimasta sconosciuta, può raggiungere tutti i dispositivi esistenti.
D'altra parte questo sistema rende sempre più forte la competizione e più alta la difficoltà di creare qualcosa di realmente innovativo rispetto a quello che si trova già sul mercato.

1.3 Applicazioni simili già esistenti sul mercato

Attualmente esistono migliaia di applicazioni per smartphone scaricabili gratuitamente o a pagamento dalle piattaforme di distribuzione digitale.
Tra le tante attualmente in commercio, le tipologie di applicazioni che più si avvicinano a quella di cui si parla in questo documento sono quelle dedicate alla geolocalizzazione dell'utente e alla cosiddetta *Social Location*, sebbene differiscano per alcune particolarità.
Oltre a questi due generi si è deciso di includere anche un'applicazione per la navigazione come *Waze* poiché, nonostante abbia uno scopo differente dall'applicazione qui descritta, condivide con essa la socialità e l'invio di informazioni anonime.
L'applicazione da me progettata combina assieme caratteristiche in comune

con tutte queste tipologie anche se si è cercato poi di realizzare un prodotto unico ed innovativo.
Naturalmente il mercato delle applicazioni per dispositivi mobili è in tale ascesa e si aggiorna così velocemente che non è possibile essere al corrente di tutte le applicazioni che ogni giorno vengono messe a disposizioni sui vari store digitali.
Resta comunque il fatto che, con questo progetto, si è cercato di creare qualcosa che, seppur simile per alcune caratteristiche a qualcosa di esistente, rappresentasse un'idea nuova.
Qui di seguito vengono descritte alcune tra le più famose ed utilizzate applicazioni appartenenti alle tipologie elencate tentando di riassumerne le relative caratteristiche nonché gli aspetti principali del loro successo.

1.3.1 Geolocalizzazione

La geolocalizzazione consiste nell'identificare la posizione geografica nel mondo reale di un oggetto, nel nostro caso di un dispositivo come un telefono cellulare o un computer.
Negli ultimi anni, a causa della crescita esponenziale degli smartphone, grandi ditte che operano a livello mondiale hanno reso disponibile la funzione di geolocalizzazione ai propri utenti.
Applicazioni di questo tipo aiutano l'utente a localizzare sé stesso su di una mappa. Infatti utilizzando il sistema GPS integrato nel dispositivo o la connessione ad internet è possibile scoprire con estrema precisione la propria posizione geografica e trovare dei servizi o delle attrazioni disponibili nelle vicinanze.
Le due applicazioni che seguono, *Mappe* e *Around Me*, sono alcune tra le più note di questo tipo per il mondo mobile.

Mappe

Mappe è un'applicazione preinstallata nell'iPhone che corrisponde alla versione mobile delle mappe di Google e permette di utilizzare tutti i servizi che queste mettono a disposizione.
Con la connessione ad internet l'utente potrà quindi vedere la propria posizione, scorrere la mappa, cercare posti di interesse nei paraggi con le relative informazioni ed ottenere il percorso migliore per raggiungerli.

Esistono delle applicazioni che si basano su questa e permettono all'utente di salvare le mappe per poterle così consultare anche off-line.
L'applicazione descritta in questo progetto di tesi utilizza proprio le mappe di Google che vengono messe a disposizione con il framework denominato Map Kit Framework.

Around Me

AroundMe permette di trovare velocemente informazioni relative al luogo in cui si trova l'utente.

AroundMe indentifica la posizione e permette di scegliere il bar, la banca, la stazione di servizio, l'ospedale, l'hotel, il cinema, il ristorante, il supermercato e i taxi vicini.
Inoltre permette di visualizzare una lista completa di tutte le attività nelle categorie scelte dall'utente con la distanza dalla sua posizione attuale.
Ogni ricerca permette di osservare la posizione su una mappa, vedere il percorso, aggiungerla ai Contatti oppure inviarla ad un altro utente via email.
La lista Qui Vicino permette di trovare informazioni da Wikipedia relative alle cose vicine al luogo dove si trova l'utente. [7]
Questa applicazione, come tante altre più e meno note, non fa altro che aggiungere servizi all'applicazione di base precedentemente descritta.
Un fattore molto importante in questo tipo di applicazioni è quello di permettere all'utente di segnalare e modificare i punti di interesse presenti nel

territorio che lo circonda e di condividerli con altri utenti. Questa caratteristica viene portata all'estremo nelle applicazioni di Social Location che andiamo ora a descrivere.

1.3.2 Social Location

Con applicazioni di Social Location si intendono quelle che combinano assieme la geolocalizzazione dell'utente alla sua appartenenza ad una comunità formata da altri utenti suoi simili, siano essi amici o perfetti sconosciuti. L'applicazione da me creata in realtà, come già detto, si discosta in parte da queste poichè, basandosi sull'anonimato dell'utente, non condivide la filosofia che sta alla base dei Social Network ma utilizza comunque la geolocalizzazione per permettere all'utente di interagire con le persone che ne condividono la posizione sul territorio.
Le applicazioni più famose e più utilizzate appartenenti a questa tipologia sono Foursquare e Facebook Places.
Facebook Places è la più recente delle due, ma essendo un'applicazione creata da Facebook, social network che conta centinaia di milioni di utenti, ha superato in breve tempo la concorrente poiché gli utenti preferiscono utilizzare un servizio a cui sono già iscritti e nel quale hanno molti contatti senza doverne aggiungere di nuovi.
Inoltre per quello che riguarda questa tipologia di applicazioni una cosa interessante è che la maggior parte della gente usa le applicazioni che utilizzano la condivisione della posizione in cambio di sconti o coupon. Non c'è nulla di strano, dal momento che il risparmio è un fattore quasi principale. [1]
Questo fa capire come non tutti gli utenti amino diffondere i propri dati in giro per il web o almeno non lo fanno senza prima avere qualcosa in cambio.

Un'altra differenza che distingue questa tipologia di applicazioni da quella da me creata è il meccanismo del check-in che utilizzano queste applicazioni. In pratica un utente per dire dove si trova effettua il check-in in un luogo (che, a volte, può essere creato dallo stesso utente). Nella mia applicazione

non esistono luoghi ma gruppi di persone che, trovandosi in posizioni molto vicine, formano le location che vengono poi visualizzate dagli altri utenti.

Foursquare

Foursquare è un'applicazione mobile e web che permette agli utenti registrati di condividere la propria posizione con i propri contatti. Il check-in nei luoghi permette di ottenere punti necessari a scalare una classifica settimanale che viene azzerata alle ore 24 di domenica, della quale fanno parte i contatti della stessa città. I check-in possono inoltre essere condivisi, insieme ad un breve status, collegando Foursquare ai propri profili Facebook e Twitter.
Quest'applicazione, come tante altre, viene inglobata dai social network per facilitare l'iscrizione e quindi l'utilizzo da parte del maggior numero possibile di utenti.
Nella versione 1.3 dell'applicazione per iPhone è stata introdotta la funzione ping che consiste nella ricezione, tramite notifiche push, degli aggiornamenti dei propri contatti. Gli utenti ricevono inoltre dei *badge*, dei riconoscimenti per aver raggiunto certi obiettivi, eseguendo il check-in in certi luoghi, ad una certa frequenza o trovandosi in una certa categoria di luoghi. Se un utente esegue il check-in in uno stesso luogo più giorni di seguito e visita un luogo più di qualsiasi altro utente nei precedenti 60 giorni ne diventa sindaco ed il suo avatar è inserito nella pagina relativa al luogo fino a quando un nuovo utente non esegue più check-in del sindaco in carica. Per diventare sindaco non valgono check-in multipli eseguiti nello stesso luogo durante lo stesso giorno. I proprietari di un'attività hanno la facoltà, qualora la pagina relativa al luogo non sia stata creata da loro, di reclamarla e di offrire sconti ed offerte al sindaco.
Gli utenti possono inoltre creare una lista privata di cose da fare e scrivere dei brevi suggerimenti per gli utenti che eseguono il check-in nel luogo stesso o in quelli vicini.
Al suo lancio nel 2009 Foursquare era disponibile in maniera limitata in sole

100 aree metropolitane in tutto il mondo. Nell gennaio 2010 il social network ha cambiato il modello alla base dei check-in permettendo di fatto di effettuarlo in qualsiasi parte del mondo. A marzo del 2010 il servizio ha superato i 500000 utenti attivi mentre a luglio 2010, in soli 4 mesi, è riuscito a superare la quota di 2 milioni. Foursquare è disponibile con applicazioni dedicate per iOS, Android, WebOS, Windows Phone 7 e Blackberry. Gli utenti Symbian e Pocket PC possono usare Foursquare attraverso Waze, disponibile anche per iOS, Android e Blackberry. Il servizio è comunque accessibile tramite browser mobili attraverso i quali i luoghi devono però essere cercati manualmente e non possono essere trovati automaticamente tramite il GPS come avviene nelle applicazioni dedicate. Il 16 aprile a Tampa, in Florida e successivamente a Manchester, New Hampshire è stato dichiarato il Foursquare Day.[6]

Come si può ben capire da questa dettagliata descrizione del funzionamento dell'applicazione, essa unisce la geolocalizzazione, alla socializzazione con altri utenti (anche tramite la condivisione sui social network) ad una specie di gioco che rende il tutto ancor più interessante.
Tutte queste caratteristiche hanno portato questa applicazione ed altre che operano nello stesso ambito ad essere scaricate ed utilizzate da migliaia di utenti in tutto il mondo.
Come già detto più volte in precedenza questa applicazione permette ai nostri amici di seguire i nostri spostamenti e i luoghi da noi maggiormente frequentati, che è proprio quello che si è cercato di evitare con il nostro progetto.

Facebook Places

Facebook Places è un'applicazione lanciata da Facebook che permette agli utenti di localizzare geograficamente la loro posizione e comunicarla ai propri amici attraverso un messaggio di status utilizzando uno smartphone.
L'applicazione permette all'utente di controllare se in quello stes-

so posto in cui si trova ci sono degli amici e chi sono.
Il tutto funziona attraverso la condivisione della propria posizione con gli utenti basandosi sul sistema dei check-in: si arriva in un posto, si apre l'applicazione di Facebook sullo smartphone, si accede alla pagina di Facebook Places e si fa check-in. Automaticamente il messaggio finirà sulla bacheca dell'utente e sulla pagina del posto in questione, un bar, un cinema, un'università, mentre l'utente sarà in grado di vedere chi altri si trova in quel momento nello stesso posto grazie alla funzione "Who are here".
Con Facebook Places si ha quindi la possibilità di incontrarsi fisicamente con gli amici in un determinato posto, oltre a poter taggare, come avviene per le foto, le persone che si trovano in quel luogo.
È possibile settare le opzioni della privacy anche per questo tipo di applicazione in modo da evitare spiacevoli "incidenti" come essere taggati da altri in diversi posti (in cui in realtà non si è presenti) o rendere pubblico a tutti dove ci si trova.

Anche per questa applicazione valgono le annotazioni scritte per la precedente.
In particolare questa applicazione è parte integrante del Social Network più grande del mondo e, mentre Foursquare punta la propria attenzione anche suoi luoghi in cui si fa il check-in grazie alle classifiche e al proprio ruolo all'interno del sistema, Facebook Places viene utilizzato prevalentemente per condividere con gli amici anche la propria posizione durante tutta la giornata, portando l'idea di rete sociale alle estreme conseguenze.

1.3.3 Social Navigation

Le applicazioni di navigazione, del tutto simili a quelle per i navigatori per auto, utilizzano il segnale GPS per ottenere la posizione dell'utente e, una volta che questo ha indicato una destinazione, fornire le indicazioni per raggiungerla passo dopo passo sotto forma di istruzioni visuali o parlate.
Il prodotto software che viene qui descritto, Waze, è una particolarità tra le

applicazioni del suo genere poiché è sì un'applicazione per la navigazione ma i cui dati derivano dalla stessa *community* degli utilizzatori. [8]
Questa tipologia di applicazioni è definita *social navigation* proprio perché combina le caratteristiche di un sistema di navigazione a quelle di un'applicazione sociale.

Waze

Waze è un'applicazione di navigazione GPS sviluppata dalla *Waze Mobile* per dispositivi mobili. Attualmente supporta iOS, Android, Windows Mobile, Symbian e BlackBerry.
Waze si discosta da tutte le altre applicazioni di navigazione per la guida poiché aggiunge il lato social, prendendo i tempi di guida degli utenti per fornire aggiornamenti real-time sul percorso e sul traffico ed inoltre permette ai guidatori di interagire tra loro inviandosi informazioni sullo stato del traffico in qualsiasi momento.
Inoltre, come nell'applicazione descritta in questa tesi, l'invio delle informazioni è anonimo. Chi contribuisce a fornire informazioni che vengono poi disposte sulla mappa non lo fa per condividere qualcosa di personale ma per aiutare gli altri, sperando che essi facciano lo stesso.
E questo è uno dei concetti su cui si torna spesso in questo lavoro di tesi.
Waze inoltre è gratuita, open-source e permette integrazioni con Foursquare, Twitter e Facebook, oltre a dei geo-giochi tra utenti che la rendono unica nel suo genere.

Capitolo 2

PRESENTAZIONE DEL PROGETTO

In questo capitolo verrà descritto il progetto e le metodologie adottate. Si proseguirà con i requisiti, la struttura con alcuni screenshot del funzionamento dell'applicazione, la presentazione delle principali problematiche sorte durante il processo di progettazione e gli approcci alla loro risoluzione.

2.1 Descrizione e metodologie adottate

Il progetto che viene qui descritto, come già accennato nell'introduzione a questo documento, consiste in un'applicazione per iPhone che utilizza la geolocalizzazione dell'utente per poter fornire dati sui luoghi che si trovano in sua prossimità frequentati in quello stesso momento da altri utenti che utilizzano la stessa applicazione.
L'idea è quella di un software leggero ed intuitivo che possa indicare all'utente quali luoghi vi sono attorno alla sua attuale posizione e quali di questi siano più o meno popolati di altri.
Come luogo non si intende necessariamente un locale fisico come può essere un bar, un ristorante, una discoteca ma anche qualcosa di non delimitato spazialmente come una piazza, un prato, una strada o un qualsiasi posto in

cui si possano incontrare le persone.
In ogni caso non è tanto importante il luogo in cui si trovano le persone quanto il gruppo di persone in sé. L'utente non vedrà indicati i nomi dei luoghi ma le coordinate nel cui intorno si trovano delle persone raggruppate per vicinanza.
Ogni utente una volta avviata l'applicazione vedrà comparire sullo schermo la mappa con lo zoom sull'area in cui si trova in quel momento grazie alla geolocalizzazione ottenuta tramite il GPS. La mappa, come quelle di Google Maps, non sarà statica ma potrà essere visualizzata dall'utente come preferisce. Quando il dispositivo avrà calcolato la posizione esatta comparirà un pin su di essa e, inviando al server le proprie coordinate assieme ad alcuni criteri di ricerca, riceverà in risposta la lista di gruppi di altri utenti che corrispondono alle caratteristiche indicate (determinato raggio, numero minimo e massimo di utenti per location).
Come già specificato precedentemente l'utente non potrà venire a conoscenza di chi si trova in un determinato luogo ma solo del numero di persone. Il server non salverà nessun dato dell'utente se non l'identificativo del suo dispositivo per poter aggiornare la posizione. È quindi garantita sia la propria privacy che quella altrui.
Per ogni location l'utente potrà venire a conoscenza di alcune informazioni statistiche come l'età media e il numero di maschi e femmine oltre ai commenti inseriti da altri utenti in modo da capire cosa sta succedendo in quel posto e decidere cosa fare di conseguenza.
L'utente stesso sarà in grado di inviare assieme agli altri dati un breve messaggio sulla location in cui si trova.
Al momento non è possibile inviare messaggi nelle altre posizioni poiché ciò potrebbe portare utenti malintenzionati ad immettere indicazioni fuorvianti su un determinato luogo; non si esclude che questa possibilità possa essere introdotta in futuro dopo aver preso le dovute precauzioni.
Infine recandosi nell'area riservata alle opzioni sarà possibile inserire i propri dati per fini statistici (età e sesso) o modificare alcune caratteristiche grafiche

dell'applicazione.

2.2 Requisiti

Il sistema che si vuole realizzare deve consentire all'utente di:

- conoscere la propria posizione e vederla visualizzata sulla mappa;
- inviare i propri dati al server;
- effettuare una ricerca di altre location indicando alcune caratteristiche;
- visualizzare le posizioni corrispondenti alle caratteristiche richieste sulla mappa;
- vedere le informazioni relative ad una location;
- scrivere un commento sulla location in cui si trova;
- modificare i propri dati;
- modificare aspetti grafici dell'applicazione;

2.3 Struttura del progetto

Dopo aver eseguito il processo di installazione dell'applicazione, essa comparirà sullo schermo del dispositivo tra le altre applicazioni, come si può vedere nella figura 2.1.

Una volta toccata l'icona relativa, l'applicazione, che nell'immagine si chiama *ProgettoTesi*, verrà avviata.
A questo punto l'utente si trova nella schermata principale in cui viene indicata la sua posizione sulla mappa attraverso un pin colorato, come si può notare nella figura 2.2. Il sistema inizialmente esegue uno zoom sull'area in

Figura 2.1: Lista delle applicazioni

cui si trova l'utente. Egli può, grazie alle caratteristiche multi-touch del dispositivo muoversi nella mappa e modificare la visualizzazione con il semplice tocco delle dita.

Figura 2.2: Schermata iniziale

Naturalmente per funzionare l'applicazione ha bisogno della posizione dell'utente che si ottiene tramite il gps del dispositivo (o con la connessione ad internet nel caso del simulatore). Essendo questo dato fondamentale, se l'applicazione per qualche motivo non riesce ad ottenerlo compare sullo schermo un messaggio di errore (come possiamo vedere nella figura 2.3) che ritorna fino a che non si riescono ad ottenere i dati geografici.
Viene messo in risalto questo errore poiché se l'utente non fornisce le proprie coordinate non può ricevere quelle delle location che lo circondano, rendendo inutile l'applicazione.

Figura 2.3: Schermata di errore

Nella barra di navigazione in alto è presente il tasto *Send Data* per inviare i propri dati e ricevere quelli delle altre location mentre nella *Tab bar* nella parte bassa dello schermo abbiamo, oltre a quello relativo alla vista attuale, un altro tasto collegato alla vista delle opzioni.

Dopo aver toccato il pulsante *Send Data* appare una nuova *view* nella quale vengono indicate le coordinate dell'utente. È poi possibile inserire un commento sulla location in cui si è attualmente e si devono indicare alcuni parametri di ricerca come il raggio, il numero minimo e quello massimo di persone delle location che si vuole cercare, come si può osservare nella figura 2.4.
Se l'utente non modifica i criteri di ricerca vengono utilizzati quelli di default.

Figura 2.4: Schermata per l'invio e la ricezione dei dati

Una volta premuto Send compare un avviso sul display che indica quante Location nel database sono state trovate, corrispondenti ai parametri immessi precedentemente. (vedi figura 2.5)

Una volta ricevuti i dati, dopo aver premuto il tasto OK per far sparire il messaggio, utilizzando il pulsante in alto a sinistra *Map* si torna alla schermata principale con la mappa in cui vengono indicate, attraverso dei pin colorati e delle circonferenze alla base, le location restituite. La larghezza del cerchio attorno al pin è direttamente proporzionale al numero di persone che risultano essere presenti in quella location mentre cliccando su di un pin

Figura 2.5: Schermata con il messaggio che indica le location trovate

appare un'annotazione che indica il numero di persone e presenta un pulsante a forma di freccia che, quando viene premuto, indirizza verso una schermata contenente le informazioni dettagliate sulla location. (vedi figura 2.6)

Figura 2.6: Location vicine all'utente

Naturalmente grazie alle funzioni messe a disposizione dal Map Kit Framework, come già detto, l'utente può muoversi nella mappa a suo piacimento scorrendo con le dita o effettuare uno zoom avanti o indietro visualizzando, in questo modo, anche i pin che si trovano leggermente più distanti risultando non visibili.

Nella schermata con le informazioni si trova l'indirizzo, la distanza dall'utente (la quale però non funziona sul simulatore ma solo sul dispositivo come indicato nelle problematiche implementative) e alcuni dati statistici quali

l'età media e il numero di uomini e donne presenti nella location. In basso, se presenti, si trovano gli ultimi commenti lasciati dagli utenti che si possono scorrere con un tocco delle dita. (vedi figura 2.7)

Figura 2.7: Schermata con le informazioni sulla location

Nella parte relativa alle opzioni l'utente può scegliere di inserire alcuni dati (al momento solo età e sesso) ed alcune caratteristiche grafiche come l'attivazione/disattivazione dei cerchi alla base dei pin ed il colore dei pin indicanti le posizioni trovate. (vedi figura 2.8)

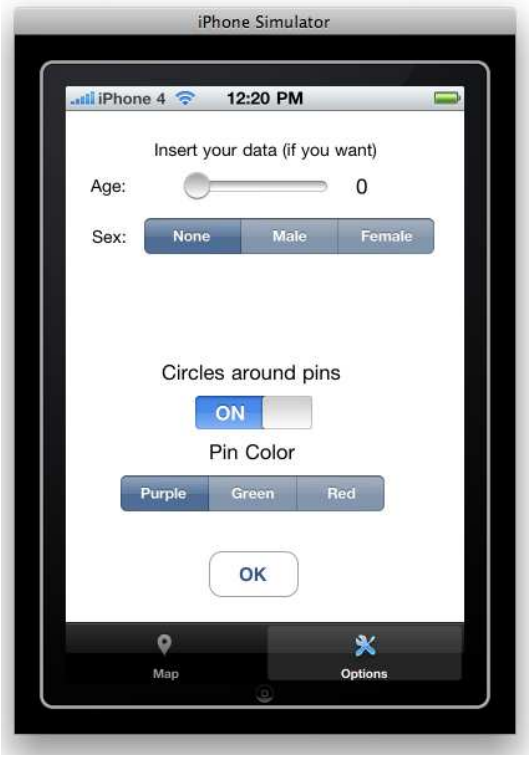

Figura 2.8: Schermata con le opzioni

2.4 Problematiche progettuali

Il primo aspetto su cui si è riflettuto è stato quello di come poter identificare l'utente all'interno del sistema cercando di evitare una registrazione al servizio e di immettere dati personali.
Si sarebbe potuto far scegliere all'utente un *nickname* ma in questo modo sarebbe stata necessaria una comunicazione con il server per vedere se il nickname scelto fosse già presente sul database e ciò avrebbe rallentato i tempi di utilizzo e reso l'applicazione più simile ad altre già disponibili sul mercato.
Un'altra soluzione avrebbe potuto essere quella di fornire all'utente al primo accesso un identificativo che, salvato sul dispositivo, sarebbe stato utilizzato per gli accessi futuri. In questo caso si sarebbe sì evitata la procedura di registrazione ma avrebbe richiesto maggior lavoro da parte del dispositivo e del server.
La scelta è caduta infine sull'utilizzo dell'udid dell'iPhone. Ogni iPhone infatti ha un *Unique Device Identifier* (UDID), che consiste in una sequenza di 40 lettere e numeri specifica per ogni dispositivo. È una specie di numero di serie ma molto più difficile da indovinare.
Naturalmente in questo caso possono sorgere dubbi sulla privacy ma poiché nel database viene salvato solo l'udid, che viene sì associato alle coordinate dell'utente ma non ai suoi dati personali, è sembrata una buona soluzione. Inoltre dopo alcune ricerche si è scoperto che diverse applicazioni, anche utilizzate da migliaia di utenti, hanno adottato proprio questo metodo.

Per quanto riguarda invece la ricezione dei dati si era pensato in un primo momento di far comunicare l'applicazione di tanto in tanto con il server in modo da ricevere automaticamente con una notifica eventuali nuove coordinate nei paraggi. Questa opzione però avrebbe richiesto un lavoro molto maggiore sia da parte del dispositivo che da parte del server e si è giunti alla conclusione che la visualizzazione di nuovi pin sarebbe dovuta avvenire solo dopo una ricerca volontaria da parte dell'utente.
Naturalmente questa soluzione, sebbene funzioni perfettamente ai nostri sco-

pi, può non essere considerata molto funzionale ed in futuro questa caratteristica potrebbe venire cambiata o migliorata.

Un'altra delle problematiche progettuali emersa sin da subito è quella riguardante la comunicazione del dispositivo con il database. Si è poi scelto di utilizzare un database PostgreSQL per le caratteristiche che vengono descritte nel prossimo capitolo ma l'accesso in remoto da dispositivo, dopo diversi tentativi è risultato di difficile implementazione anche a causa delle scarse informazioni reperibili sia nella documentazione ufficiale che in rete.
L'idea iniziale era infatti quella di far eseguire le *query SQL* direttamente al dispositivo per non sovraccaricare troppo il server, nel caso di un numero elevato di utenti connessi.
Ma per le problematiche riscontrate è stato deciso di far fare tutto il lavoro del database al server, ottenendo in questo modo anche tempi di computazione più brevi per il dispositivo.
L'iPhone semplicemente manderà una richiesta con alcuni dati al server, esso interagirà con il database e fornirà al dispositivo i risultati che verranno poi elaborati e mostrati sull'applicazione all'utente.

Un altro aspetto su cui si è ragionato prima di procedere all'implementazione è la gestione dei messaggi.
In un primo momento si pensava di far lasciare all'utente messaggi su qualsiasi location in modo da poter sia esprimere un parere in quella in cui si è sia domandare cosa stesse succedendo in un'altra vicina. Quest'idea è stata scartata poiché in questo modo un utente in malafede avrebbe l'opportunità di inserire messaggi falsi e fuorvianti sulle altre location spingendo altri utenti ad evitarle anche senza motivo.
Permettendo all'utente di inserire messaggi solo per la location in cui si trova si risolve questa problematica poiché non è egli a decidere ma il sistema automaticamente salva il messaggio inserito dall'utente nella tabella del database relativa alla location in cui l'utente si trova realmente.

Infine ci si è chiesti come implementare l'interfaccia per renderla più intuitiva ed usabile possibile. Si è così deciso di utilizzare poche finestre importanti e di disporre una barra di navigazione nella parte alta dello schermo ed una di scelta nella parte bassa.
Ad esempio le caratteristiche dell'utente come età e genere compaiono nella vista delle opzioni nonostante vengano inviati assieme agli altri dati dalla vista per l'invio e la ricezione dei dati. Questa scelta è stata dettata dal fatto che questi dati, al contrario di altri, sono opzionali e non vengono modificati ogni volta come invece succede per i criteri di ricerca.
Se in futuro dovessero essere aggiunte funzionalità che richiedano nuove viste sarà facile aggiungerle a quelle già presenti.

Capitolo 3

IMPLEMENTAZIONE

Qui di seguito viene descritta l'implementazione del progetto a partire dagli strumenti utilizzati con una breve descrizione e la ragione di tale scelta. Verranno delineati successivamente in dettaglio tutti gli aspetti implementativi sia per la parte client che per quella server.
Nell'ultima parte sono elencate le problematiche riscontrate durante l'implementazione del progetto e il metodo che è stato scelto per la loro risoluzione. Si concluderà con i test effettuati ed i risultati che questi hanno generato.

Essendomi avvicinato alla programmazione di software in ambiente Mac con questa tesi, prima di procedere all'implementazione e dopo aver progettato gli aspetti teorici per poter realizzare l'applicazione, mi sono documentato sui metodi migliori di muovere i primi passi in questo ambiente ed, in particolare, nella programmazione per dispositivi mobili come iPhone.
A questo proposito è stato molto utile il libro, edito da Apress, *Beginning iPhone Development* [9] grazie al quale ho creato passo passo diverse applicazioni per iOS per poter poi sfruttare al meglio le conoscenze acquisite in questo progetto di tesi.

3.1 Descrizione dei software e delle tecnologie utilizzati e ragioni della scelta

3.1.1 Ambiente di sviluppo

Xcode

Xcode è un ambiente di sviluppo integrato (*Integrated development environment*, IDE) sviluppato da Apple Inc. per agevolare lo sviluppo di software per Mac OS X e iOS. È fornito gratuitamente in bundle con il sistema operativo a partire da Mac OS X 10.3 Panther, sebbene sia in grado di generare programmi per qualsiasi versione di Mac OS X. Estende e rimpiazza il precedente tool di sviluppo della Apple, *Project Builder*, che era stato ereditato dalla NeXT. Ufficialmente Xcode non funziona su Mac OS X 10.2 Jaguar.
Xcode lavora in congiunzione con Interface Builder (anch'esso proveniente da NeXT) che permette agli sviluppatori che usano Carbon e Cocoa di disegnare interfacce grafiche per le applicazioni usando uno strumento grafico, senza la necessità di scrivere decine di righe di codice. L'interfaccia risultante è salvata in un file .nib o, nelle versioni più recenti come file .xib.
Xcode include GCC, che è in grado di compilare codice C, C++, Objective C/C++ e Java. Supporta ovviamente i framework Cocoa e Carbon, oltre ad altri.
Una delle caratteristiche tecnologicamente più avanzate di Xcode è che supporta la distribuzione in rete del lavoro di compilazione. Usando Bonjour e Xgrid è in grado di compilare un progetto su più computer riducendo i tempi. Supporta la compilazione incrementale, è in grado di compilare il codice mentre viene scritto, in modo da ridurre il tempo di compilazione.

Dalla versione 3.1, Xcode è anche lo strumento per sviluppare le applicazioni native per iPhone e iPod touch.
Dalla versione 3.2, inoltre, è anche possibile sviluppare applicazioni per iPad. [10]

Nel marzo del 2011 è stata rilasciata la versione 4.0 la cui caratteristica più importante è la nuova interfaccia grafica che integra tutte le funzionalità all'interno di un'unica finestra, compreso interface builder con i relativi *property ispector*. Un'altra novità introdotta con questa versione è la funzionalità Fixit che fornisce alcuni suggerimenti per il completamento del codice e l'individuazione di alcuni tra i più comuni bug o errori di battitura sottolineandoli con una linea rossa.
Rende inoltre più semplice il merge del codice permettendo di comparare diverse versioni dello stesso file.
Infine Xcode 4 per la prima volta è stato reso disponibile a pagamento anche tramite la distribuzione digitale attraverso il Mac App Store.

Nonostante i miglioramenti apportati nell'ultima versione di questo prodotto, per il progetto è stata utilizzata la versione 3.2.3 poiché, quando è stata progettata ed implementata questa applicazione, la nuova versione non era ancora disponibile e si è deciso di non aggiornare l'ambiente di sviluppo ad applicazione ultimata per evitare eventuali problematiche.
Xcode è stato utilizzato per la realizzazione di tutto il codice relativo all'applicazione mentre con Interface Builder è stata curata la creazione dell'interfaccia.
È stato scelto questo ambiente di sviluppo poiché è lo strumento ideale per realizzare applicazioni in ambiente Mac, essendo stato sviluppato dalla Apple stessa. È risultato essere molto intuitivo e veloce sia durante la fase di scrittura che di realizzazione dell'interfaccia sia per quanto riguarda il debug. Inoltre, mettendo a disposizione un simulatore, è stato utilizzato anche nella fase di test dell'applicazione.

Il funzionamento dell'ambiente di sviluppo è molto intuitivo.
Una volta creato un nuovo progetto e scelta la tipologia tra i template disponibili, appare l'area di lavoro.
Nella colonna a sinistra si ha la lista dei file che compongono il progetto suddivisi in cartelle. Nel caso di quest'applicazione avendo scelto inizialmente un'applicazione universale (che può funzionare cioé sia su iPhone che su iPad) è stata creata una cartella che contiene le classi specifiche per l'iPhone, una per l'iPad ed una *Shared* che contiene le classi che sono comuni ad entrambi i dispositivi. Poiché in un secondo momento si è deciso di realizzare l'applicazione solo per iPhone per le ragioni precedentemente descritte, la maggiorparte delle nostre classi è contenuta nella cartella iPhone. Abbiamo poi una cartella *Other Sources* che contiene i file .xib di interface builder in cui sono state disegnate le interfacce ed una cartella *Frameworks* che contiene appunto i framework inclusi nel progetto.
Nella parte centrale dell'ambiente di sviluppo è possibile vedere il codice della pagina selezionata mentre nella barra in alto sono presenti i pulsanti per il debug e la compilazione.

Per quanto riguarda Interface Builder esso viene utilizzato per disegnare le interfacce risparmiando così al programmatore la scrittura di grandi quantità di linee di codice.
Quando viene aperto un file .xib compaiono quattro nuove finestre (solo nelle versioni di Xcode precedenti la 4.0 con la quale è stato unito tutto in un'unica finestra): una contiene la vista selezionata, un'altra mette a disposizione un elenco di oggetti che possono essere trascinati sulla vista, una terza permette di ispezionare i vari oggetti e modificarne caratteristiche e collegamenti mentre l'ultima mostra gli elementi che compongono la vista, come è facilmente osservabile nella figura 3.2.

Infine Xcode mette anche a disposizione, attraverso *Instruments* alcune ap-

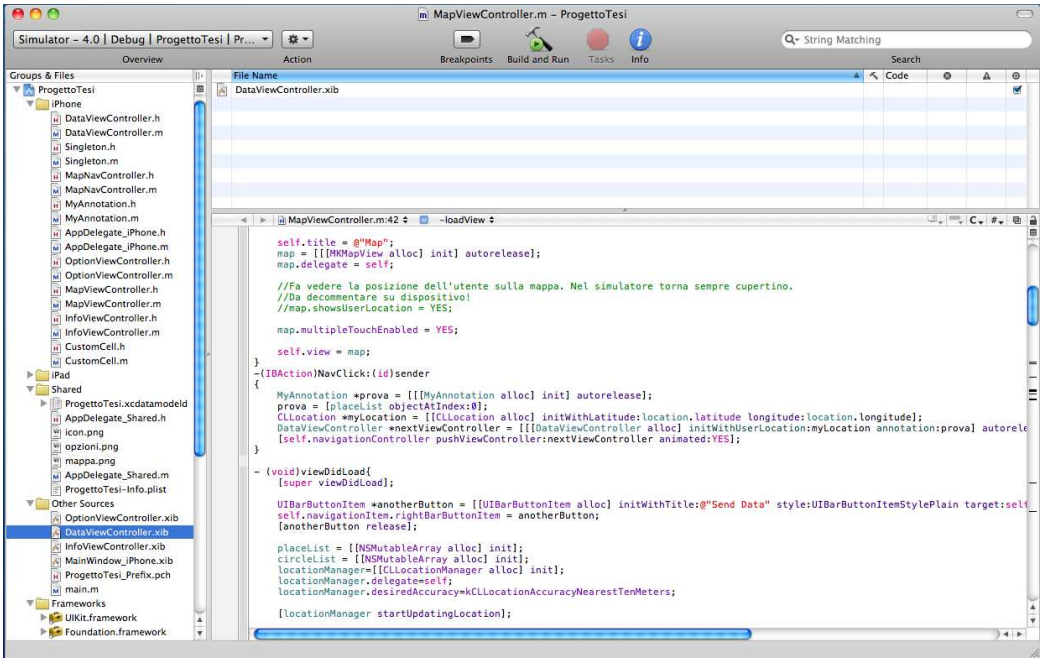

Figura 3.1: Ambiente di sviluppo Xcode

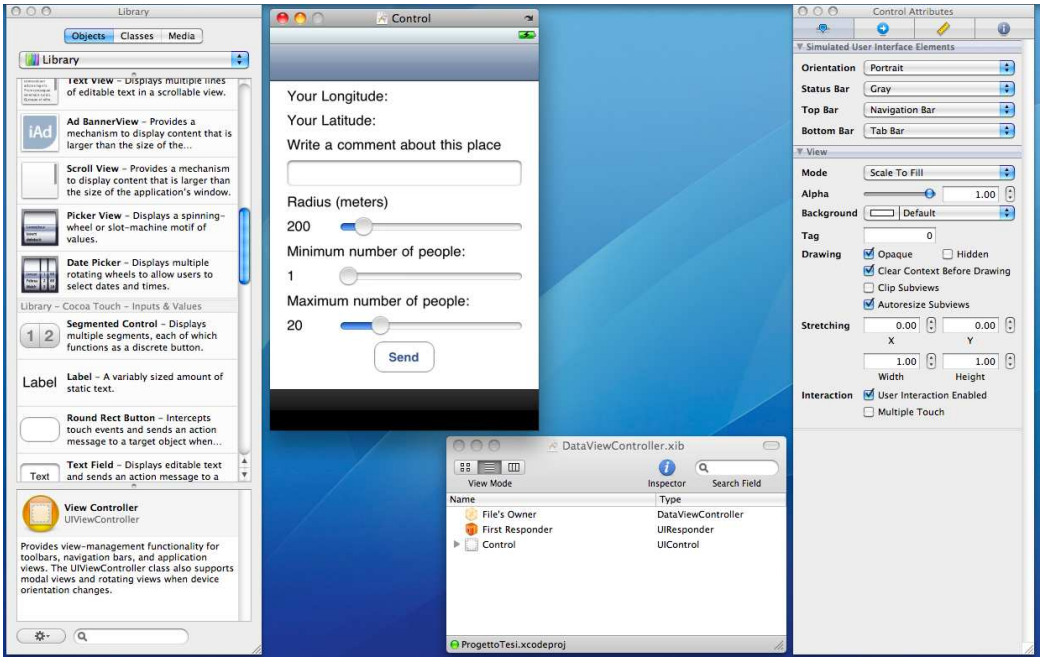

Figura 3.2: Finestre di Interface Builder

plicazioni per il monitoraggio del funzionamento del software.
Per utilizzare questi strumenti basta scegliere dal menù in alto *Run>Run with Performance Tool* dove compare poi un menù che permette la scelta tra i diversi applicativi utilizzabili.
In particolare nella fase di test è stata utilizzato lo strumento *Leaks* che permette di vedere, durante il funzionamento dell'applicazione, se ci sono blocchi di memoria che vengono allocati e non rilasciati o altre "perdite" di memoria. Correggendo in base ai suggerimenti il codice, si rende il software più veloce e funzionale, riducendo lo spreco di memoria.

3.1.2 Database

PostgreSQL

PostgreSQL è unanimemente riconosciuto come il miglior ORDBMS (*Object Relational DataBase Management System*) non commerciale presente ad oggi nel panorama mondiale.
Questo sistema, rilasciato con licenza open source, è un'alternativa ad altri prodotti sia liberi come MySQL, Firebird SQL e MaxDB sia quelli a codice chiuso come Oracle, Informix o DB2 ed offre caratteristiche uniche nel suo genere che lo pongono per alcuni aspetti all'avanguardia nel settore dei database.
Fra le caratteristiche principali di PostgreSQL val la pena dire che fornisce supporto per le sub-queries, le chiavi esterne (Foreign Keys), i Triggers, le View oltre ai tipi di dati geometrici e dati relativi alle reti.
Ha più di 15 anni di sviluppo attivo alle spalle ed un'architettura comprovata che gli ha fatto guadagnare una buona reputazione per quello che riguarda

l'affidabilità, l'integrità dei dati e la correttezza.
Può essere utilizzato su tutti i maggiori sistemi operativi, inclusi Linux, UNIX (e di conseguenza anche MAC OS X), e Windows.

PostGIS

PostGIS è un'estensione spaziale per il Database Management System PostgreSQL distribuito con licenza GPL.
Fornisce i tipi di dati specificati negli standard dell'*Open Geospatial Consortium*. In particolare è un geodatabase e fornisce il sistema di gestione dati sui quali è basato un GIS (*Geographic(al) Information System*). [14]

Il database PostgreSQL è stato installato sulla macchina utilizzata come server per gestire i dati relativi alle location ed ai dispositivi.
La scelta è ricaduta su questo database sia per la licenza libera sia perché è l'unico a gestire dati geometrici che, assieme all'utilissima estensione PostGIS, permettono di utilizzare dati geospaziali come la longitudine e la latitudine ed eseguire funzioni già implementate, evitando di far eseguire operazioni complesse come, ad esempio, il calcolo delle distanze, al dispositivo.

pgAdmin

La piattaforma pgAdmin è Open Source e viene utilizzata per l'amministrazione e la gestione di database PostgreSQL.
Può essere utilizzata su piattaforme Linux, FreeBSD, Solaris, Mac OS X e Windows per gestire PostgreSQL dalla versione 7.3 in poi su qualsiasi piattaforma.
L'interfaccia grafica di pgAdmin supporta tutte le caratteristiche di PostgreSQL e rende l'amministrazione più semplice permettendo di effettuare query sql, mostrare le tabelle e gestire direttamente i database.

Per queste sue caratteristiche è stata installata sulla macchina per poter gestire facilmente i database, partendo dalla creazione degli stessi fino alla fase

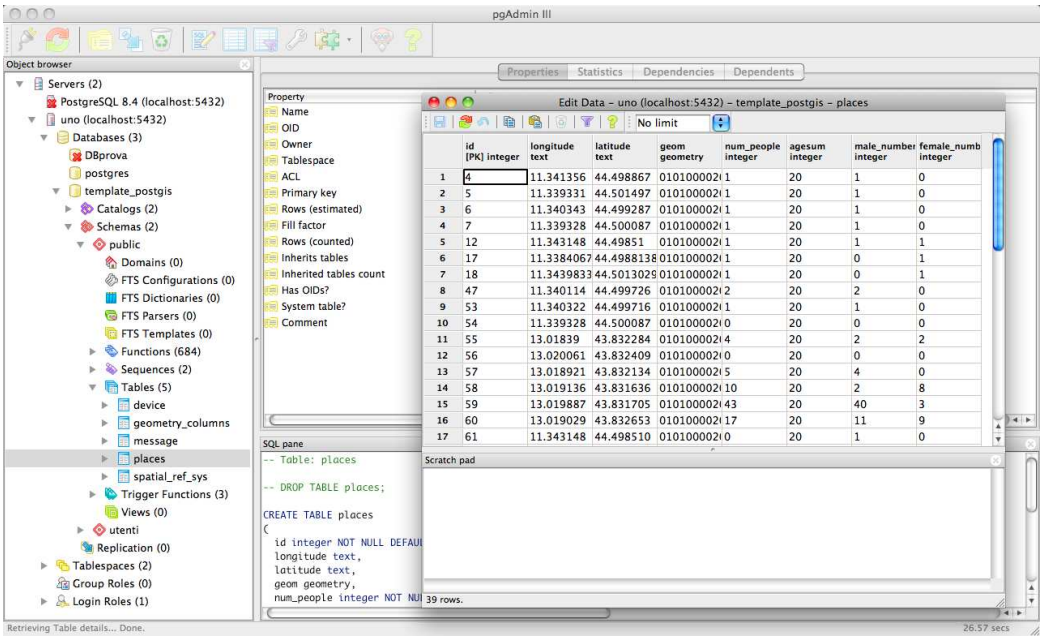

Figura 3.3: Interfaccia di pgAdmin III

iniziale di test in cui sono stati inseriti all'interno delle varie tabelle dati fittizi per simulare la presenza di utenti sul territorio.
Per eseguire test più realistici si è poi deciso di trasferire il database su di uno spazio web. Per non doverlo ricreare da zero è bastato semplicemente esportare con pgAdmin le tabelle del database creato in locale ed importarle tramite query SQL sulla macchina server, dove era stato attivato precedentemente un account di PostgreSQL con PostGIS.

3.1.3 Webserver

PHP

PHP (acronimo ricorsivo di PHP: Hypertext Preprocessor, preprocessore di ipertesti) è un linguaggio di scripting interpretato, con licenza open source e libera (ma incompatibile con la GPL), originariamente concepito per la programmazione Web e quindi la realizzazione di pagine web dinamiche.

Attualmente è utilizzato principalmente per sviluppare applicazioni web lato server ma può essere usato anche per scrivere script a riga di comando o applicazioni standalone con interfaccia grafica.
L'elaborazione di codice PHP sul server produce codice HTML da inviare al browser dell'utente che ne fa richiesta. Il vantaggio dell'uso di PHP e degli altri linguaggi Web come ASP e .NET rispetto al classico HTML deriva dalle differenze profonde che sussistono tra Web dinamico e Web statico.
Un esempio di software scritto in php è MediaWiki, su cui si basano progetti wiki come Wikipedia e Wikizionario. [11]

Il php viene utilizzato nella parte server del progetto per creare le pagine richiamate dal dispositivo per interagire con il database inviando e ricevendo dati.
È stato scelto questo linguaggio non tanto per le sue performance rispetto ad altri linguaggi quanto perché oltre ad essere open source, era già stato da me adottato in progetti passati.

3.2 Principali aspetti implementativi

3.2.1 Librerie esterne

Core Location framework

Il Core Location framework permette di determinare la posizione corrente o la direzione associate ad un dispositivo. Questo framework usa l'hardware disponibile per determinare la posizione dell'utente e la sua direzione. Le classi e i protocolli contenuti in questo framework possono essere usati per

configurare e stabilire la consegna di eventi relativi alla posizione e alla direzione. Può anche essere utilizzato per definire regioni geografiche e monitorare quando l'utente attraversa i confini di queste regioni. [16]

In particolare nell'applicazione questo framework è stato utilizzato per determinare la posizione dell'utente e per gestire tutte le coordinate, ricevute come risultato della ricerca nel database, e le operazioni tra esse sulla mappa.

Map Kit framework

Il Map Kit framework fornisce un'interfaccia per includere le mappe direttamente nelle finestre e nelle viste. Questo framework inoltre fornisce supporto per le annotazioni sulla mappa, l'aggiunta di *overlays* e permette di eseguire ricerche attraverso il *reverse geocoding* per determinare informazioni data una specifica coordinata. [17]

Le funzioni appartenenti a questo framework gestiscono la mappa che viene visualizzata oltre a tutte le possibilità di interazione con essa e permettono di ottenere, date le coordinate geografiche, l'indirizzo che viene visualizzato nell'Info view.
Inoltre queste funzioni consentono di gestire tutti i pin oltre alle annotazioni che compaiono quando essi vengono selezionati. Grazie agli overlay vengono disegnati i cerchi alla base dei pin.

3.2.2 Principali classi dell'applicazione

DataViewController

Questa classe che estende *UIViewController* gestisce la vista per l'invio e la ricezione di dati. Al suo interno sono implementate le label delle coordinate, un campo di testo per l'inserimento di un commento e gli slider relativi alle opzioni quali raggio, numero minimo e massimo di utenti. Inoltre si prende cura dell'invio dei dati al webserver e della ricezione di essi, nonché

delle operazioni di parsing dell'Xml.
Quando nella classe *MapViewController* viene premuto il bottone di navigazione in alto a destra, essa inizializza questa classe fornendo le coordinate geografiche dell'utente.
Questi dati, insieme a quelli inseriti tramite l'interfaccia dall'utente, vengono salvati sotto forma di una stringa e poi inviati al server via HTTP, utilizzando il metodo *POST*, verso la pagina *sendData.php*. Il server utilizza i dati, interroga il database e fornisce in risposta una stringa Xml. In caso il dispositivo non riesca a leggere i dati viene mostrata una finestra di errore altrimenti la stringa viene parsata utilizzando la classe *NSXMLParser* ed i singoli dati inseriti in un array globale e resi disponibili alle altri classi per la visualizzazione sulla mappa.

Per quanto riguarda l'invio dei dati in questa classe è indicato l'indirizzo del server locale nell'istruzione sottostante.

Listing 3.1: Url del server

```
NSURL *url = [NSURL URLWithString:@"http://localhost:8888/Tesi/sendData.php"];
```

Questo indirizzo dovrà essere sostituito con quello della macchina server esterna al momento di implementare l'applicazione su dispositivo.

Singleton

Nella programmazione ad oggetti un *singleton* è un design pattern creazionale che ha come scopo quello di garantire che di una determinata classe venga creata una e una sola istanza, e di fornire un punto di accesso globale a tale istanza.
Nel nostro caso si occupa di gestire variabili globali che possono essere scritte e lette dalle diverse classi che compongono l'applicazione. Ad esempio nella classe *DataViewController* quando i dati Xml sono stati parsati attraverso la funzione *impostaArray* l'array con le location diventa globale e può in que-

sto modo essere letto anche dalle altre classi grazie alla funzione *ritornaArray*.

Listing 3.2: Array globale

```
// Con questo metodo imposto il valore dell'item condiviso
+ (void)impostaArray:(NSMutableArray *)valore{
        arrayCondiviso = valore;
}

// Con questo metodo, prelevo il valore dell'item condiviso
// da qualsiasi classe
+ (NSMutableArray *)ritornaArray{
        return arrayCondiviso;
}
```

Le altre funzioni di questa classe vengono utilizzate per impostare e ritornare variabili globali come il colore dei pin, l'attivazione/disattivazione degli overlay attorno ai pin e l'array dei messaggi della location selezionata per poterli visualizzare nella vista delle informazioni.

MyAnnotation

La classe MyAnnotation, che estende NSObject ed implementa il protocollo di MKAnnotation, è utilizzata per creare un'annotazione personalizzata che comparirà sulla mappa nel momento in cui l'utente toccherà il pin.
Contiene le coordinate dell'utente oltre a tutti i dati della location cui si riferisce oltre ad un titolo ed un sottotitolo.
Viene creata un'istanza di MyAnnotation per ogni location sulla mappa all'interno del metodo *viewWillAppear* nella classe *MapViewController* in modo che la lista delle annotazioni sia aggiornata ogni volta che viene ricaricata la vista con la mappa.
In particolare il metodo *initWithCoordinate* permette di inizializzare l'annotation partendo dalle coordinate del punto e da tutti gli altri dati (numero di persone, età, utenti maschi e femmine, id dell'annotazione). In questo caso il titolo verrà impostato a "Altra location" ed il sottotitolo mostrerà il numero di utenti in quella location.

Diversamente il metodo *initWithUserLocation* crea un'annotazione con le coordinate dell'utente. Quest'ultimo metodo, una volta installata l'applicazione sul dispositivo, potrà non essere più utilizzato poiché sul dispositivo, al posto del pin che indica la location dell'utente avremo un pallino azzurro come indicato nelle problematiche e nella descrizione della classe *MapViewController*.

OptionViewController

Classe del tutto simile a DataViewController che gestisce la vista per le opzioni e viene richiamata cliccando sul bottone a destra della *tab bar* che si trova nella parte bassa dello schermo.
L'interfaccia corrispondente contiene uno slider per l'età dell'utente, un bottone per la scelta del sesso, un bottone per impostare il colore dei pin e ed uno *switch* per permettere all'utente di abilitare o disabilitare i cerchi alla base dei pin (overlay) che servono ad indicare il numero di utenti.
Una volta impostate le opzioni (non necessariamente tutte) queste vengono rese globali attraverso la classe singleton descritta precedentemente in modo da non doverle reinserire ogni volta che si cambia schermata.
I dati come l'età e il sesso vengono poi inviati dalla classe *DataViewController* al server mentre il colore del pin così come gli overlay vengono impostati nella classe *MapViewController*.

MapViewController

Classe principale del progetto, estende anch'essa UIViewController, ed implementa MKMapViewDelegate e CLLocationManagerDelegate, utilizzate per la gestione della mappa e delle coordinate geografiche. Si occupa di gestire e visualizzare la posizione dell'utente all'interno della mappa, dello zoom e dello scorrimento di essa e del posizionamento dei nuovi pin ottenuti dalla richiesta al webserver.

Quando la vista viene caricata inizialmente questa classe gestisce lo zoom

e lo scorrimento della mappa per poter mostrare la posizione esatta dell'utente.
Ogni volta che invece vengono ricevuti i dati e, dalla vista *DataViewController* si clicca sul pulsante Map nella barra di navigazione per tornare a questa vista, fa un refresh facendo comparire i pin delle nuove location trovate.
Per far ciò nel metodo *viewWillAppear*, che viene chiamato ogni volta che si ricarica la vista, viene svuotato l'array delle location, attraverso il singleton viene letto l'array aggiornato e per ogni *annotation* in esso viene inserito un pin e, grazie alla funzione *addOverlay* un cerchio che indica gli utenti alla sua base.
Il raggio del cerchio ha come dimensione il doppio degli utenti più uno.
I cerchi alla base dei pin possono essere disabilitati dalla finestra delle opzioni in modo da rendere la consultazione più veloce e meno confusionaria nel caso di un numero elevato di location.
Come si può vedere in questo estratto del codice che si trova all'interno di un ciclo for, per ogni elemento all'interno dell'array *webPlaceList* viene creata una nuova annotazione a cui vengono passati tutti i dati relativi a quella location. L'annotazione viene poi aggiunta all'array *placeList* e rilasciata. Nell'ultima riga viene calcolato il raggio del cerchio che verrà visualizzato attorno al pin.

Listing 3.3: Inserimento dell'annotazione e calcolo del raggio

```
MyAnnotation *newAnnotation = [[MyAnnotation alloc]
        initWithCoordinate:locationNew
        numberOfPeople:peopNumber
        averageAge:avgAge
        male:maleN
        female:femaleN
        idOfAnnotation:annotationID
        ];

[placeList addObject:newAnnotation];
[newAnnotation release];

int CircleRadius = peopNumber*2 + 1;
```

Infine attraverso la funzione *[locationManager startUpdatingLocation];* viene aggiornata la mappa.
Dopo aver rappresentato sullo schermo le posizioni delle nuove location con i relativi pin in questa classe vengono gestite le annotazioni su cui è indicato il numero di utenti e il collegamento con la vista delle informazioni relative ad una location, fornito dal metodo *calloutAccessoryControlTapped* che viene richiamato quando l'utente preme il pulsante a forma di freccia che si trova sull'annotazione.

È importante mettere in evidenza, come anche descritto successivamente nella sezione relativa alle problematiche, che quando l'applicazione verrà installata su dispositivo sarà opportuno decommentare la seguente istruzione che permette di indicare la posizione esatta dell'utente sulla mappa attraverso un pallino azzurro.

Listing 3.4: Funzione per mostrare la posizione dell'utente sulla mappa

```
//map.showsUserLocation = YES;
```

Una volta che la posizione dell'utente viene visualizzata grazie a questa istruzione, naturalmente si potrà anche rimuovere dal codice la parte relativa all'annotazione sulla posizione dell'utente che si trova nel metodo *didUpdateToLocation*.

InfoViewController

Altra classe che estende UIViewController e gestisce la vista delle informazioni relative ad una specifica location.
Ha al suo interno delle UILabel che contengono i dati di quella posizione come la distanza dalla posizione in cui si trova l'utente (che però non viene calcolata nel simulatore), il numero di utenti in quella posizione, la media dell'età e il numero di utenti maschi e femmine. In basso, quando presenti, sono mostrati gli ultimi messaggi lasciati dagli utenti inseriti in una tabella scorrevole.

Viene richiamata quando si preme il *callout button* all'interno dell'annotazione relativa ad un pin.
Nel metodo principale nel codice di questa classe vengono inizialmente lette le informazioni relative alla location specificata attraverso i singleton, come già descritto in questa sezione.
In particolare l'indirizzo viene ottenuto attraverso un'operazione di *reverse geocoding* partendo dalla longitudine e latitudine del punto, utilizzando le apposite funzioni messe a disposizione dal Map Kit Framework.
Nell'estratto di codice che segue viene mostrata la funzione per ottenere il reverse geocoding. In particolare si è deciso di mostrare soltanto il nome della via nell'indirizzo della location poiché, esaminando le location attorno all'utente, sembrava inutile inserire altri dati come la città, la regione o lo stato.
Il secondo metodo restituisce un errore quando, per qualche motivo, l'operazione non va a buon fine.

Listing 3.5: Funzione per ricavare l'indirizzo attraverso il reverse geocoding

```
-(void)reverseGeocoder:(MKReverseGeocoder *)geocoder
        didFindPlacemark:(MKPlacemark *)placemark{
                NSDictionary *addressData = placemark.addressDictionary;
                CGFloat y = 50;
                int count = 0;
                for (NSString *key in [addressData allKeys]) {
                        if ([key compare:@"FormattedAddressLines"]
                        != NSOrderedSame) {
                                count++;
                                if([key isEqualToString:@"Street"])
                                {
                                        addressLabel.text = [NSString
                                          stringWithFormat:@"%@: %@",
                                          key,
                                          [addressData objectForKey:key]];
                                }
                                y += 24;
                        }
                }
}

- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder
```

```
        didFailWithError:(NSError *)error{
        addressLabel.text = @"Reverse geocoding failed, try again";
}
```

Successivamente, come avviene all'interno della classe *DataViewController* viene fatta una richiesta HTTP di tipo POST indicando l'id della location alla pagina *getMessage.php* sul server che restituisce (se presenti) i messaggi relativi con una stringa Xml che viene poi parsata.
I messaggi sono così salvati in un array ed inseriti in una tabella per essere mostrati nella vista.
Anche in questo caso l'indirizzo del server locale è stato sostituito al momento di testare l'applicazione con un server remoto via web.

3.2.3 File relativi al server

I seguenti file sono tutti scritti in php, si trovano all'interno del webserver e vengono richiamati dall'applicazione per poter interagire in scrittura e/o lettura con il database.

pg_conn

Questo file contiene i dati per la connessione al database e viene incluso, attraverso l'istruzione *require('pg_conn.php');* all'interno degli altri file del webserver.
Nella funzione seguente vengono indicati il nome del database, il nome dell'utente, l'host e la porta.

Listing 3.6: Connessione al database

```
$conn = @pg_connect('dbname=template_postgis
        user=postgres
        host=localhost
        port=5432');
```

In caso di trasferimento su di un altro server occorrerà fare attenzione alla modifica di questi dati per far sì che i dispositivi possano continuare a colle-

garsi al database. In particolare per il trasferimento su uno spazio web sarà necessario inserire anche il campo *password*.

sendData

L'applicazione, attraverso il file DataViewController, esegue una richiesta HTTP di tipo POST su questo file inviando l'udid dell'utente, la sua posizione, il timestamp, il sesso e l'età (se impostati nella finestra delle opzioni) e il messaggio (se inserito dall'utente) oltre alle caratteristiche delle location richieste dall'utente come raggio e minimo e massimo numero di utenti.
Questo file gestisce l'inserimento o la modifica dei dati all'interno del database ed interroga lo stesso fornendo in risposta al dispositivo una stringa Xml con le posizioni risultanti dalla ricerca.

Per prima cosa controlla se l'udid dell'utente è già presente nel database.
Se c'è aggiorna la data mentre il sesso e l'età solo se sono presenti entrambi e non erano già stati inseriti. Poi controlla se il dispositivo è ancora a 10 metri dalla location in cui si trovava precedentemente, in caso positivo lascia così com'è o inserisce un messaggio se l'utente l'ha inserito, altrimenti decrementa di un'unità il numero di utenti in quella location e controlla se ce n'è una nuova nell'intorno di 10 metri.
Se ne trova una incrementa il numero di utenti, modifica i dati statistici ed inserisce il messaggio aggiornando il riferimento alla location nella tabella dei dispositivi.
Nell'altro caso viene aggiunta al database una nuova location che ha come coordinate quelle dell'utente.

Quando invece il dispositivo non è presente nel database, per prima cosa inserisce l'udid assieme agli altri dati nella tabella *device* e poi si comporta a seconda dei casi come descritto precedentemente.

Questo per quello che riguarda l'invio dei dati.

Per la ricezione seleziona id e coordinate inferiori ad un certo raggio e con un determinato numero di persone, come indicato dall'utente, direttamente come stringa Xml che viene poi restituita e parsata dall'applicazione.

Listing 3.7: Query per la ricerca delle location vicine

```
if(!$query = @pg_query("SELECT query_to_xml('
   SELECT id,longitude,latitude,num_people,agesum,male_number,female_number
   FROM places
   WHERE distance(
      transform(
         PointFromText(
            \'POINT($longitude $latitude)\', 4269),32661),
            geom) < $radius
   AND num_people
   BETWEEN $minPeop AND $maxPeop',false,false,'')"))
```

In questo estratto di codice viene utilizzata la funzione *query_to_xml* che restituisce i risultati della query direttamente come una stringa Xml. Nella query più interna vengono richieste tutte le location che distano meno del raggio indicato dalla posizione dell'utente e il cui numero di persone è compreso tra il minimo e il massimo.
La funzione *distance* che permette di calcolare la distanza tra due coordinate geografiche fa parte di PostGIS.

getMessage

Questo file viene chiamato dalla vista delle informazioni su di una specifica location con l'id di quest'ultima ed interroga il database restituendo (se presenti) i messaggi relativi alla posizione indicata. Viene utilizzato lo stesso metodo indicato nel codice precedente cambiando la query con quella che segue.

Listing 3.8: Query per ottenere i messaggi relativi ad una location

```
if(!$query = @pg_query("SELECT query_to_xml('
   SELECT id,text
   FROM message
   WHERE location_id = $id
   ORDER BY id DESC LIMIT 10',false,false,'')"))
```

pg_close_conn

Questo file, che viene incluso in fondo agli altri sempre attraverso la funzione *require*, contiene semplicemente una funzione per terminare la connessione al database.
Non è obbligatorio chiudere la connessione ma è comunque consigliato farlo.

3.2.4 Tabelle del database

Come già spiegato nella sezione relativa agli aspetti implementativi, è stato utilizzato un database postgreSQL con l'estensione postGIS per gestire le operazioni con le coordinate geospaziali.

device

In questa tabella vengono salvati i dati relativi ai dispositivi che inviano i propri dati al server.
Per ogni dispositivo si hanno: l'udid (*Unique Device Identifier*) del dispositivo come identificativo, il timestamp con la zona geografica dell'ultimo invio dei dati da parte di quel dispositivo nel formato *'yyyy-mm-dd hh:mi:ss+02'*, il collegamento all'id della location in cui si trova l'utente in quel momento, l'età e il sesso dell'utente (se indicati) come si può vedere nella tabella seguente.

Tabella 3.1: Tabella device

did [*PK*] **uuid**	**time** **timestamp with time**	**people_id** **integer**	**age** **integer**	**sex** **integer**
9c72(...)	2011-06-20 16:37:43+02	57	40	2

Il campo did è stato tagliato per ragioni di spazio.
Nel campo sex l'intero 1 indica genere maschile, 2 femminile.

places

La tabella places contiene le informazioni sulle location.
Ogni *entry* contiene come chiave primaria un identificativo che incrementa automaticamente, la longitudine e latitudine del punto, un campo di tipo geometry, il numero di utenti in quella posizione, la somma delle età degli utenti (per calcolare velocemente la media), il numero di maschi e quello di femmine (per le statistiche). Si osservi un esempio nella tabella qui sotto.

Tabella 3.2: Tabella places

id [*PK*] **integer**	**long** **text**	**lat** **text**	**geom** **geometry**	**people** **integer**	**agesum** **integer**	**male_n** **integer**	**female_n** **integer**
57	13.018921	43.832134	0101(...)	5	80	4	1

Il campo geometry viene ricavato trasformando le coordinate del punto da quelle di un sistema sferico ad un altro (come si può meglio capire in [12]) in modo da essere poi utilizzato con postGIS per effettuare operazioni sui dati spaziali.
Nella tabella l'elemento del campo geometry è stato tagliato per ragioni di spazio.

message

Contiene tutti i messaggi lasciati dagli utenti riguardo una specifica location.
Ha un campo identificativo che viene incrementato automaticamente, uno testuale e la chiave esterna relativa alla location cui fa riferimento.

Tabella 3.3: Tabella message

text **text**	**id** [*PK*] **integer**	**location_id** **integer**
Questa è la prova di un messaggio per questa location.	2	57

geometry_columns e spatial_ref_sys

Queste due tabelle sono state generate per poter lavorare con i dati geografici.
In particolare la tabella *spatial_ref_sys* fa parte di PostGIS e contiene più di 3000 sistemi di riferimento spaziali noti e i relativi dettagli mentre la tabella *geometry_columns* viene utilizzata per poter lavorare con i dati di tipo geometry ed è definita come si può vedere in questa tabella.

Tabella 3.4: Tabella geometry_columns

oid	**f_table _catalog** [*PK*] **char**	**f_table _schema** [*PK*] **char**	**f_table _name** [*PK*] **char**	**f_geometry _column** [*PK*] **char**	**coord _dimension integer**	**srid integer**	**type character**
24604	''	public	people	geom	2	32661	GEOM

3.3 Funzionamento dell'applicazione

Quest'applicazione è stata creata per funzionare su iPhone e (con le opportune modifiche al codice) su iPad.
Come già accennato in precedenza non può funzionare su dispositivi sprovvisti di GPS come iPod Touch poiché necessita del GPS per la localizzazione dell'utente attraverso le sue coordinate.
Per poter funzionare l'applicazione dovrà essere scaricata dall'App Store o installata direttamente sul dispositivo; la sua icona comparirà così tra le altre applicazioni nel menù dello smartphone.

Una volta avviata, attraverso alcune funzioni del *Core Location framework*, ottiene la geolocalizzazione del dispositivo prima facendo uno zoom della mappa (ottenuta tramite il *Map Kit Framework*) nell'area in cui si trova l'utente e poi segnalando attraverso un pin la posizione esatta dell'utente.
È necessario puntualizzare anche in questa sezione che, una volta trasportata su dispositivo, dopo aver effettuato le dovute modifiche al codice precedentemente descritte, la posizione dell'utente non verrà indicata tramite un pin

ma tramite un pallino azzurro.
Per lo zoom sono stati fissati i parametri *latitudeDelta* e *longitudeDelta* a 0.003 ottenendo un'area di qualche centinaio di metri attorno all'utente.
Naturalmente l'utente potrà grazie alle funzionalità messe a disposizione dal Map Kit framework fare zoom avanti o indietro oppure scorrere la mappa in qualsiasi momento.
In alto nella finestra c'è una barra di navigazione. Toccando il pulsante *Send* a destra l'utente cambia vista passando a quella per l'invio e la ricezione dei dati.
In questa finestra vengono visualizzate la longitudine e la latitudine dell'utente calcolate precedentemente e sono presenti: un campo per l'inserimento di un breve commento sulla *location* in cui si è, la scelta del raggio, attorno a cui cercare altre posizioni con utenti, con un range che varia da 15 a 2000 metri, il minimo e il massimo numero di persone che si vogliono avere in queste posizioni, compresi tra 1 e 100. Naturalmente il numero minimo di utenti deve sempre essere minore/uguale rispetto quello massimo.
In questo modo l'utente può scegliere luoghi più o meno affollati che si trovano ad una determinata distanza dalla sua posizione attuale a seconda di quello che sta cercando.
Una volta premuto il pulsante *Send* vengono inviati al server sia l'identificativo del dispositivo (udid) che la posizione dell'utente indicata come gradi di longitudine e latitudine, oltre agli altri dati impostati dall'utente.
L'udid del dispositivo, essendo unico e non modificabile, permette di evitare l'iscrizione al servizio utilizzando un nickname o dei dati personali.
Il webserver salva nel database l'udid del dispositivo se è la prima volta che questo invia dati oppure aggiorna l'ora dell'invio, la posizione ed i dati statistici (sesso ed età) che possono essere indicati nella vista delle opzioni. Inoltre viene aggiornata la tabella delle location incrementando il numero di persone di quella dove si trova l'utente e decrementando quella in cui era prima se egli si è spostato ed era già presente nel database o creandone una nuova se non ce n'è nessuna nell'intorno di 10 metri dalla posizione attuale dell'utente.

Quando una location raggiunge quota 0 utenti, non viene più mostrata tra i risultati.

Ricapitolando, per evitare di creare troppe location, non ne viene creata una per ogni utente ma ne viene creata una nuova soltanto se nel raggio di 10 metri dall'utente non ce ne sono. In questo modo la mappa risulta meno confusionaria e anche il lavoro diminuisce sia per il dispositivo (che deve mostrare sullo schermo meno pin) sia per il server (che deve calcolare meno risultati).

In caso l'utente abbia riempito il campo del messaggio relativo alla location in cui egli si trova, questo verrà inserito nella tabella dei messaggi relativi alla sua location sul database senza però associarlo in alcun modo all'utente che l'ha inviato, rendendolo così anonimo.
Attraverso i dati inseriti dall'utente viene interrogato il database richiedendo le location che corrispondono alle caratteristiche indicate (distanza, numero di persone) che vengono restituite come dati Xml al dispositivo.
Per calcolare la distanza tra le coordinate dell'utente e quelle delle location salvate sul database è stato molto utile postGIS che, lavorando sui dati geografici, ha funzioni create appositamente che permettono di calcolare le distanze tra coordinate geografiche in tempi brevissimi.
Attraverso il pulsante di navigazione in alto a sinistra l'utente potrà tornare alla visualizzazione della mappa.
I dati ricevuti, parsati dal dispositivo vengono disposti, come pin, sulla mappa. Ogni pin avrà alla base un cerchio, ottenuto tramite apposite funzioni del Map Kit framework, che indica il numero di persone in quella posizione. Maggiore è il cerchio maggiore sarà il numero degli utenti.
Questi cerchi, ottenuti tramite gli overlay del Map Kit Framework, possono essere disabilitati dalla schermata delle opzioni se causano rallentamenti e cali di prestazioni o rendono la mappa troppo confusionaria nel caso di un grande numero di pin visualizzati sullo schermo.

Selezionando con un tocco il pin di una location, si aprirà un'annotazione indicante il numero di utenti e, premendo ulteriormente il pulsante a forma di freccia all'interno dell'annotazione si passerà alla vista informativa con tutti i dati relativi a quel luogo.
Una volta aperta la nuova vista comparirà un messaggio che indica che i dati sono stati ricevuti con successo.
Tra le informazioni, oltre all'indirizzo ottenuto tramite il reverse geocoding e la distanza dalla posizione dell'utente (non disponibile sul simulatore) sono presenti anche dati statistici come il numero di utenti, l'età media, il genere degli utenti e i commenti che questi hanno rilasciato su quel particolare luogo.
A volte, almeno sul simulatore, il reverse geocoding risulta un po' lento e la prima volta che si apre la schermata informativa sulla location al posto dell'indirizzo compare una scritta che indica che il reverse geocoding è fallito. Il problema spesso è risolvibile semplicemente tornando alla mappa per poi ricaricare nuovamente la finestra della location vedendo comparire così l'indirizzo corretto.
Nel caso il problema persista significa che ci sono degli errori durante le operazioni di geocoding inverso o a causa della mancanza di informazioni o a causa dello stesso segnale GPS.
I commenti degli utenti sono inseriti nelle celle di una tabella e, nel caso siano molti, possono essere scorsi con un movimento delle dita sul touch-screen.
Infine, utilizzando il menù nella barra in basso, l'utente può accedere alla vista delle opzioni dove ha la possibilità di inserire i propri dati utili alle statistiche dei luoghi (sesso ed età) che vengono inviati assieme agli altri dati e cambiare alcune caratteristiche grafiche dell'interfaccia come il colore dei pin, e la visualizzazione dei cerchi che indicano il numero di utenti alla loro base.

3.4 Problematiche implementative riscontrate e risoluzione

Le maggiori difficoltà si sono presentate durante la fase di testing e debugging dell'applicazione.
Come viene più dettagliatamente descritto nella sezione successiva, l'applicazione è stata testata soltanto sul simulatore fornito da Xcode e non su dispositivi reali.
Questa cosa ha causato alcune problematiche: l'istruzione

```
map.showsUserLocation = YES;
```

presente nella classe *MapViewController* serve ad indicare sulla mappa automaticamente la posizione dell'utente. Quest'ultima però, nel simulatore, è indicata sempre come la sede della Apple a Cupertino, in California (come si può vedere nella figura 3.4).

Si è quindi deciso, per poter continuare a sviluppare e testare l'applicazione, di commentare quell'istruzione e creare un pin con la relativa annotazione per la posizione reale dell'utente. Così, attraverso le funzioni del Core Location Framework, vengono calcolate la longitudine e latitudine (nel caso del simulatore che non ha il GPS ricavate dalla connessione ad internet del laptop) e viene inizializzata un'annotazione particolare che indica la posizione dell'utente sulla mappa.
Sempre questo problema rende vano l'utilizzo della funzione *distanceFromLocation* fornita dal Core Location che calcola la distanza tra una posizione e quella dell'utente. Infatti avviando l'applicazione da simulatore, nella vista delle informazioni relative ad una specifica location, il campo della distanza è sempre 0.
Questo problema si può facilmente risolvere decommentando l'istruzione precedentemente citata al momento di installare l'applicazione sul dispositivo.

Per quanto riguarda le location nel database un problema era quello del-

Figura 3.4: Posizione dell'utente sul simulatore

l'aggiornamento del numero di utenti e dei relativi dati statistici.
Poiché si è scelto di rendere facoltativo l'inserimento dei dati statistici (età e genere) le varie medie calcolate dividendo la somma dei dati inseriti per gli utenti in una location spesso tornavano risultati sbagliati. Questo poiché si è simulato che non tutti gli utenti scelgono di inserire i propri dati. Si è risolto il problema immettendo nel database età e sesso dell'utente soltanto quando vengono inseriti entrambi. In questo modo, per calcolare la media di età, è bastato sommare le età degli utenti in una location e dividere il risultato per il numero di utenti che hanno inserito i dati, ottenuto sommando i maschi alle femmine (quando questi sono diversi da 0).

Una problematica che non è stata risolta modificando il codice è quella relativa ai cerchi attorno ai pin.
Essi servono ad indicare il numero di persone che sono in una determinata location ma, almeno nel simulatore, compaiono molto lentamente, rendendo difficile la consultazione. Questo problema si presenta soltanto con il simulatore dell'iPhone 4, non con quello della versione precedente dell'iPhone dove gli overlay compaiono sì lentamente ma in tempi accettabili.
Questi cerchi vengono creati grazie agli overlay forniti dal Map Kit Framework e non si trovano segnalate problematiche simili sul web né tra le reference del framework. Si pensa quindi che possa essere un problema del rendimento grafico sul simulatore e che venga risolto semplicemente portando l'applicazione su dispositivo.
Per sicurezza è stato inserito uno *switch* nella vista relativa alle opzioni che permette di abilitare/disabilitare i cerchi attorno ai pin. Questa possibilità può risultare utile anche se il problema non si dovesse presentare sul dispositivo poiché con molte location sulla mappa i cerchi potrebbero rendere la visualizzazione più confusionaria.

3.5 Test effettuati

Xcode contiene un simulatore che permette di testare le applicazioni create su iPhone (vecchi modelli), iPhone4 e iPad.

I test dell'applicazione qui descritta sono stati tutti effettuati sul simulatore

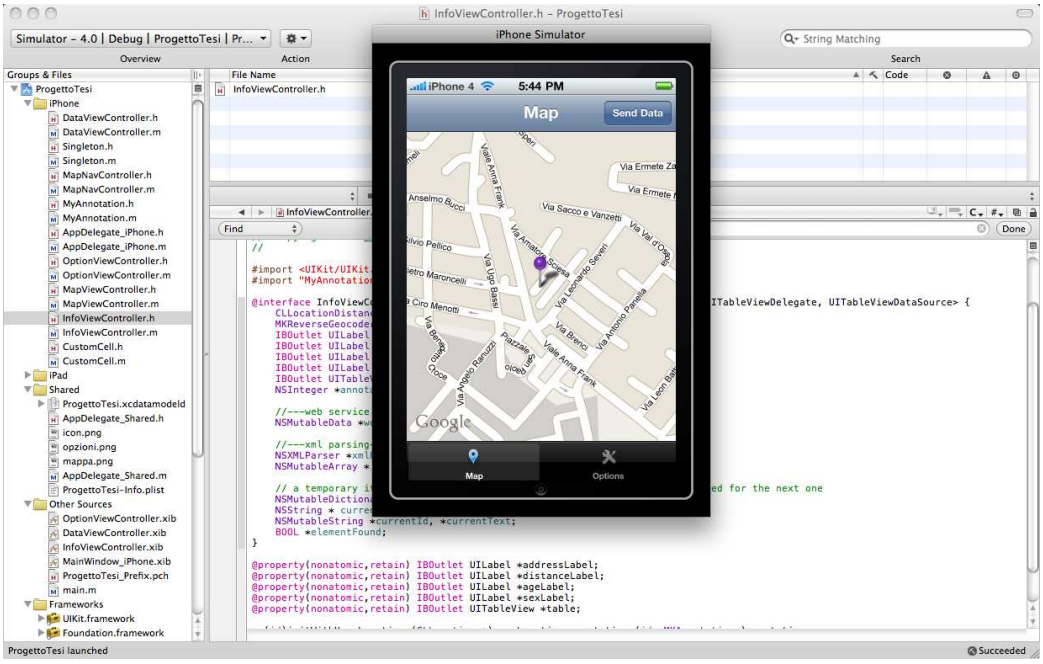

Figura 3.5: Simulatore di Xcode

non permettendo l'Apple l'installazione su dispositivo senza un account da sviluppatore (a pagamento).

Per quanto riguarda la parte server invece, il webserver con il database PostgreSQL è stato installato inizialmente in locale sulla stessa macchina utilizzando la piattaforma MAMP (acronimo che si riferisce ad un set di programmi liberi comunemente utilizzati insieme per far girare un sito web dinamico sul sistema operativo Mac OS X) che fornisce un webserver *Apache* ed il supporto al PHP.

Il server locale è stato utilizzato durante la fase di implementazione e debug

dell'applicazione, in modo da avere tempi di risposta più rapidi e poter procedere nel lavoro anche in mancanza della connessione ad internet.
Per la fase di test invece il database è stato esportato attraverso *pgAdmin* ed importato, tramite query SQL, su di un servizio di hosting gratuito fornito da *Alwaysdata*. [18]
L'hosting gratuito mette a disposizione un quantitativo ridotto di spazio su disco e traffico mensile ma supporta comunque diversi linguaggi tra cui PHP oltre a diversi database tra i quali c'è PostgreSQL con l'estensione PostGIS e permette la gestione via web con *phpPgAdmin*.
Questo servizio è stato utilissimo per la fase di test poiché, grazie ad esso, è stato possibile osservare la risposta dell'applicazione e le relative tempistiche nel comunicare con un server remoto. Eseguendo le query sql in locale infatti, i tempi sono molto brevi sia quando ci sono pochi dati in risposta sia quando ce ne sono molti di più, rendendo i test in locale pressoché inutili.
Dopo aver importato il database sull'host del sito si è proceduto a modificare le due linee di codice nelle classi *DataViewController* ed *InfoViewController* già descritte nella parte relativa all'implementazione, sostituendo l'indirizzo locale con il nuovo indirizzo remoto.
Si è provveduto inoltre ad aggiornare la funzione di connessione al database, contenuta nella pagina *pg_conn.php*, con i dati per l'accesso al servizio per poi uploadarla via ftp, assieme alle altre, sullo spazio web messo a disposizione.

Per simulare delle *location* attorno all'utente, esse sono state inserite manualmente nel database, dopo aver osservato le coordinate di luoghi vicini su Google maps. In questo modo è stato possibile fingere che attorno all'utente ci fossero diverse *location* con un numero variabile di utenti.

Naturalmente le prestazioni del simulatore non possono essere paragonate a quelle che si otterrebbero con l'applicazione installata su dispositivo infatti, a seconda delle singole funzioni che compongono il programma, il funzionamento dell'applicazione sul simulatore può risultare più lento o più veloce.

Per quanto concerne il server invece i dati ottenuti risultano realistici essendo stato installato in remoto ed accedendovi tramite web.
In ogni caso i test qui descritti servono a rendere al meglio l'idea di quello che fa l'applicazione e di quale sarebbe il suo funzionamento una volta installata su dispositivo.

3.5.1 Funzionamento generale

Per testare il funzionamento generale dell'applicazione sono stati inseriti manualmente dei dati nel database.
Si è utilizzata un'applicazione web che restituisce le coordinate di un punto selezionato sulla mappa per trovare la longitudine e la latitudine di alcune location attorno alla posizione dell'utente in quel momento (in particolare si è utilizzata la posizione del laptop su cui è installato Xcode ricavata grazie alla connessione ad internet). Si è cercato di inserire dati abbastanza eterogenei sia a livello di disposizione sulla mappa che come numero di persone, media di età e genere.
Si è poi avviata l'applicazione diverse volte cercando di utilizzarne tutte le funzioni simulando il comportamento di un utente che si trova in giro col proprio dispositivo mobile.

Con 10 location inserite nell'area che circonda l'utente, l'applicazione ha funzionato senza problemi, i risultati della ricerca sono stati restituiti molto velocemente ed i pin posizionati correttamente così come le informazioni relative alle varie location.
L'unico elemento negativo è il rallentamento delle prestazioni al momento di disegnare i cerchi attorno ai pin delle location ma di ciò se ne è parlato dettagliatamente nel capitolo relativo alle problematiche di implementazione.

3.5.2 Scalabilità

Il termine scalabilità in informatica, e in altre discipline, si riferisce, senza scendere troppo nello specifico, alla capacità di un sistema di crescere o decrescere (aumentare o diminuire di scala) in funzione delle necessità e delle disponibilità. Un sistema che gode di questa proprietà viene detto scalabile. Per vedere se la nostra applicazione è scalabile si è deciso di far visualizzare di volta in volta un numero maggiore di location attorno all'utente.
Naturalmente, si guarda se il sistema è scalabile dal punto di vista del singolo dispositivo ovvero se l'applicazione riesce a sostenere un numero sempre maggiore di dati da leggere e mostrare all'utente. Sarebbe opportuno anche testare il funzionamento del server all'aumentare degli utenti che vi si collegano ma questo non è stato possibile poiché, non avendo la possibilità di testare l'applicazione su dispositivo, bisognerebbe utilizzare un simulatore per ciascuno degli ipotetici utenti.
Ci si è quindi limitati ad aumentare il numero location e per far ciò sono state inserite manualmente con il metodo già descritto decine di location con le coordinate all'interno di una circonferenza di raggio di 2km attorno all'utente (distanza che è stata poi impostata come massima nella schermata di invio dei dati).
Sono state poi introdotte delle funzioni nel codice per calcolare il tempo impiegato dal programma a restituire i risultati. Il tempo viene preso al momento in cui viene premuto il pulsante Send e poi nel momento in cui finisce il parsing dei risultati utilizzando la funzione *CFAbsoluteTimeGetCurrent()*.
Si sono poi fatte delle ricerche prima con un raggio di 50 metri, poi 100, poi 200 ed infine 2000 in modo da avere di volta in volta sempre più pin presenti sulla mappa. Per quanto riguarda il numero minimo e massimo di utenti per location questi sono stati posti al minimo (1) e massimo (100) in modo da ricevere in output il maggior numero di pin possibile.
Per eseguire questi test sono stati disattivati i cerchi attorno ai pin mediante l'apposito switch nella finestra delle opzioni.
Il sistema ha continuato a funzionare anche con oltre 20 pin sulla mappa

senza bloccarsi o avere un calo di prestazioni. Nella tabella sottostante sono indicati i risultati assieme ai tempi medi in secondi.

Tabella 3.5: Risultati test sulla scalabilità

raggio	min_p	max_p	location trovate	tempo (sec)
50	1	100	1	0.259
100	1	100	4	0.290
200	1	100	11	0.328
500	1	100	24	0.341
1000	1	100	26	0.348

Come si vede nella tabella i tempi di risposta aumentano all'aumentare del numero di location restituite ma è molto importante sottolineare come più aumenta il numero di location trovate, minore è l'incremento del tempo. Ad esempio la differenza di tempo tra 1 e 4 location trovate è inferiore di qualche centesimo di secondo a quella tra 11 e 24.
Questo dato ci da un buon segnale riguardo la scalabilità e cioè che probabilmente anche continuando ad aumentare il numero di location che vengono restituite dal database, i tempi di risposta saranno sempre più vicini tra loro rendendo l'applicazione utilizzabile senza problemi anche con grandi quantitativi di dati.
Va anche fatto notare che in una location vengono raggruppati gli utenti nel raggio di 10 metri dal punto centrale di essa e quindi il numero di location attorno al raggio massimo (2000 metri) dall'utente non potrà mai superare una certa soglia.
Dal test e da queste considerazioni si può supporre che anche con il massimo numero possibile di location attorno all'utente i tempi di risposta saranno attorno ad un secondo rendendo l'applicazione comunque performante.

Sul simulatore, durante i test, si è notato che, anche con gli stessi dati, a volte il numero di location restituite variava leggermente poiché le coordinate dell'utente non erano sempre le stesse.

Questo è dovuto al metodo che il simulatore utilizza per ottenere la posizione tramite la connessione ad internet e non si verifica sul dispositivo. Per non falsare i test si sono presi in considerazione tutti i risultati con le stesse coordinate dell'utente.

Capitolo 4

CONCLUSIONI E SVILUPPI FUTURI

4.1 Conclusioni

In questa tesi si è cercato di sviluppare un'applicazione per dispositivi mobili che consenta la localizzazione di gruppi di persone nell'ambiente circostante in grado di rispondere alle esigenze di efficienza e usabilità dell'utente.

Inizialmente è stato svolto un lavoro di ricerca sulle tecnologie ed i software già esistenti sul mercato nel campo dei dispositivi mobili per cercare di capire quale fosse la situazione attuale e come si potesse creare qualcosa utile agli utenti e allo stesso tempo originale.
Sono stati poi inquadrati e valutati i tanti strumenti disponibili per adempiere allo scopo e la scelta è caduta sull'ambiente di sviluppo Apple, in particolare per il del dispositivo iPhone, e l'utilizzo del Map Kit Framework per la gestione delle mappe mentre, a livello server, postgreSQL con la sua estensione postGIS come database per contenere i dati e PHP come linguaggio per le pagine web che prendono in consegna i dati e li elaborano.
L'ambiente Apple è stato scelto combinando disponibilità dei mezzi, curiosità

e guardando anche l'andamento del mercato dei dispositivi che hanno come sistema operativo iOS come iPhone ed iPad.
Come prima esperienza di progettazione ed implementazione sia in ambiente Apple che per dispositivi mobili la realizzazione di quest'applicazione è stata anche una sfida personale ma i risultati ottenuti sono stati soddisfacenti e i mezzi utilizzati molto utili al raggiungimento di un buon risultato.
L'applicazione che è venuta fuori dal lavoro di progettazione prima ed implementazione poi risponde alle funzionalità che erano state poste inizialmente e può essere il punto di inizio per futuri ampliamenti o nuovi progetti basati sulle idee qui presentate.
Naturalmente un'applicazione del genere per poter adempiere correttamente al proprio scopo ed essere efficace per chi l'utilizza ha bisogno di un alto numero di utenti che dispongano di un dispositivo Apple ed utilizzino contemporaneamente questa applicazione. Si può però pensare di sviluppare la stessa applicazione anche in altri ambienti per dispositivi diversi in modo che sempre più utenti possano trarne vantaggio.
Inoltre, come già detto più volte, l'applicazione per poter funzionare correttamente ha bisogno di ulteriori test, della prova su dispositivo e quindi sul campo. Infatti i test che sono stati eseguiti ci hanno permesso di testarne il corretto funzionamento, la scalabilità e l'usabilità da parte dell'utente di cui però non si può essere completamente sicuri se non dando l'applicazione in mano ad utenti veri che si muovono nel loro ambiente.

In ogni caso, oltre agli aspetti meramente tecnici ed implementativi adottati per la realizzazione di questo progetto, restano molto importanti alcuni concetti che si sono voluti trasmettere, come ad esempio quello che creare un'applicazione che aiuti la socialità non debba necessariamente andare contro la privacy dei suoi utenti mettendo in mostra i loro dati personali ma possa semplicemente aiutarli a destreggiarsi nel mondo che li circonda.
L'utente, soprattutto nelle applicazioni che ne riguardano la socialità, non deve più venir visto come singolo ma sentirsi parte della comunità ed aiutarla

in ogni aspetto della vita quotidiana aiutando così anche se stesso.
In questo modo le applicazioni software per dispositivi mobili, oltre ad essere utilizzate per svago o per passare il tempo, possono diventare molto utili semplificando per molti aspetti la vita quotidiana delle persone che le utilizzano.

4.2 Sviluppi futuri

L'applicazione creata permette numerose modifiche ed ampliamenti sia per migliorarne la resa e la stabilità sia per aggiungere nuove potenzialità e caratteristiche.
Infatti, per limiti di tempo e competenze, con questo progetto non si è potuta creare un'applicazione perfetta e definitiva nel suo campo ma è stato comunque realizzato qualcosa che è sì completo e utile ma che da allo stesso tempo un input a chi voglia cimentarsi in questo stesso campo.

Tra gli sviluppi futuri che vengono subito in mente e che è possibile realizzare con poche modifiche ai sorgenti del programma si potrebbe implementare, ad esempio, un sistema di messaggistica tra utenti che si trovano in location diverse per scambiarsi in tempo reale pareri e opinioni o semplicemente per comunicare. Per far ciò è sufficiente mettere a punto una specie di forum, con domande e risposte nelle schede delle location o qualcosa di più diretto, come può essere un servizio chat, magari senza passare dal server ma utilizzando un metodo di comunicazione diretta tra i dispositivi di due o più utenti.

Sarebbe utile realizzare un sistema di rating delle location e di descrizione dell'evento a cui si assiste o del posto in cui ci si trova per dar modo all'utente di scegliere più facilmente come muoversi. Allo stesso fine si potrebbe anche pensare di dare la possibilità all'utente di caricare sulla scheda informativa di una location una foto che lì è stata scattata direttamente con il dispositivo.

Per quanto riguarda i dati che vengono inseriti dall'utente, nell'applicazione è possibile inserire solo età e sesso mentre potrebbe essere utile rendere possibile l'inserimento di altri dati, sempre restando anonimi, come la professione e gli interessi in modo da realizzare una ricerca più selettiva.
Naturalmente ogni nuova informazione che contribuisca a dare maggiori elementi decisionali all'utente contribuirà a rendere l'applicazione più completa.

Un'altra possibilità potrebbe consistere nel permettere all'utente di effettuare una ricerca non solo attorno alle proprie coordinate ma anche fornendo delle coordinate a sua scelta.
L'utente manderebbe comunque automaticamente le proprie coordinate al server per aggiornare i dati globali, ma specificando un indirizzo o delle coordinate geografiche al momento della ricerca potrebbe ottenere le location che si trovano in un luogo più distante, magari dove ha intenzione di andare.

Si potrebbe poi pensare di integrare l'applicazione con i più famosi social network ma sempre garantendo all'utente l'anonimato, per non stravolgere il senso dell'applicazione stessa.
Una soluzione potrebbe essere ad esempio quella di colorare con un colore diverso i pin delle location in cui si trovano degli individui che l'utente ha come contatti sul social network, in modo da distinguere quelle location in cui sono presenti persone che potenzialmente potrebbe conoscere sempre senza indicarne l'identità.

Passando alla parte server, una volta che l'applicazione abbia cominciato la propria distribuzione, è facile pensare alla realizzazione di un sito web che permetta la semplice consultazione delle varie location sparse in giro per il mondo in modo da permettere anche a chi non dispone di un dispositivo mobile di vedere cosa succede attorno a sé e di partecipare alla comunità.
In questo caso però verrebbe a mancare la formula dell'invio di dati propri per ricevere quelli degli altri utenti. Limitatamente a questo caso si potrebbe

pensare ad una registrazione al solo servizio web ponendo così tutti gli utenti sullo stesso livello.

Oltre agli sviluppi futuri descritti in queste pagine se ne possono pensare tanti altri che rendano l'applicazione sempre più completa.
Resta comunque il fatto che per la velocità con cui si evolvono i dispositivi, i sistemi operativi ed il mercato, anche una volta che questo software dovesse venire rilasciato e distribuito, avrebbe comunque bisogno di aggiornamenti e ampliamenti periodici per garantire agli utenti un utilizzo sempre privo di problemi.

Bibliografia

[1] Why Mobile Users Aren't Checking In, 2011, *http://mashable.com/2011/05/04/social-location-apps-study/*.

[2] iPhone - Wikipedia, 2011, *http://it.wikipedia.org/wiki/Iphone*.

[3] Berg: Smartphone shipments grew 74% in 2010, 2011, *http://www.bgr.com/2011/03/10/berg-smartphone-shipments-grew-74-in-2010/*.

[4] IDC Forecasts Nearly 183 Billion Annual Mobile App Downloads by 2015: Monetization Challenges Driving Business Model Evolution, 2011, *http://www.idc.com/getdoc.jsp?containerId=prUS22917111*.

[5] App Store - Wikipedia, 2011, *http://it.wikipedia.org/wiki/App_Store*.

[6] Foursquare - Wikipedia, 2011, *http://it.wikipedia.org/wiki/Foursquare*.

[7] AroundMe per iPhone, iPod touch e iPad nell'iTunes App Store, 2011, *http://itunes.apple.com/it/app/aroundme/id290051590?mt=8*

[8] Waze GPS & traffico - sociale, divertimento! per iPhone, iPod touch e iPad nell'iTunes App Store, 2011, *http://itunes.apple.com/it/app/waze-gps-traffico-sociale/id323229106?mt=8*.

[9] Dave Mark, Jeff LaMarche, *Beginning iPhone 3 Development: Exploring the iPhone SDK*, Apress, 2010.

[10] Xcode - Wikipedia, 2011, *http://it.wikipedia.org/wiki/Xcode*.

[11] PHP - Wikipedia, 2011, *http://it.wikipedia.org/wiki/Php*.

[12] Using POSTGIS to find points of interest within a radius, 2007, *http://unserializableone.blogspot.com/2007/02/using-postgis-to-find-points-of.html*.

[13] PostgreSQL, 2011, *http://www.postgresql.org/*.

[14] PostGIS - Wikipedia, 2011, *http://it.wikipedia.org/wiki/PostGIS*.

[15] pgAdmin, 2011, *http://www.pgadmin.org/*.

[16] Core Location Framework Reference, 2010, *http://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html%23//apple_ref/doc/uid/TP40007123*.

[17] Map Kit Framework Reference, 2010, *http://developer.apple.com/library/ios/#documentation/MapKit/Reference/MapKit_Framework_Reference/_index.html*.

[18] The Ultimate Web Hosting — alwaysdata, 2011, *http://www.alwaysdata.com/*.