

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

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

Implementazione di un Software Mobile per l'Organizzazione di Allenamenti in Gruppo nel Ciclismo

Relazione finale in

Mobile Web Design

Relatore
Dott. Mirko Ravaioli

Presentata da
Christian Gamberini

Sessione 2

Anno Accademico 2012/2013

INDICE

PREFAZIONE	7
1 INTRODUZIONE	9
1.1 Il Ciclismo ed i suoi Allenamenti	9
1.2 I Social Network	10
1.3 Le Potenzialità degli Smartphone	11
2 FASE DI PROGETTAZIONE	13
2.1 Analisi del Domino.....	13
2.1.1 Analisi della memorizzazione utente	13
2.1.2 Analisi della memorizzazione allenamenti.....	14
2.1.3 Funzionalità dell'applicazione	14
2.2 Progettazione delle Entità.....	16
2.2.1 Prima individuazione delle entità.....	16
2.2.2 Analisi delle entità individuate.....	18
2.2.3 Individuazione delle entità definitive	20
2.2.4 Studio delle relazioni fra le entità.....	22
2.3 Studio delle Varie Funzionalità Interne.....	27
2.3.1 Funzionalità di ricerca	27
2.3.2 Funzionalità lato server	28
2.4 Progettazione Estetica.....	28
2.4.1 Scelta dei colori	29
2.4.2 Presentazione dell'icona.....	29
2.4.3 Descrizione dei vari layout	29
2.5 Conclusione Fase di Progettazione	36
3 FASE DI IMPLEMENTAZIONE	37
3.1 Conversione delle Tabelle del Modello Relazionale	37
3.1.1 Tabella Allenamento	37
3.1.2 Tabella Utente	38
3.1.3 Tabella Luogo.....	39
3.1.4 Tabella Commento	39

3.1.5	Tabella Partecipa.....	39
3.1.6	Tabella Passa_Da.....	40
3.2	Scelta del Sistema di Implementazione.....	40
3.3	Introduzione agli Strumenti Applicativi.....	40
3.3.1	Le activity Android.....	40
3.3.2	Ascoltatori XML.....	41
3.3.3	I Dialog.....	41
3.3.4	Shared Preferences.....	42
3.3.5	AsyncTask.....	42
3.4	Implementazione delle classi in Java.....	43
3.5	Il Server.....	44
3.5.1	Estensione del protocollo di comunicazione.....	45
3.5.2	Elenco delle funzioni in linguaggio Php.....	45
3.5.3	Classe server in Java.....	49
3.6	Procedure di Login e Registrazione.....	50
3.6.1	Login automatico.....	50
3.6.2	Login manuale.....	51
3.6.3	Registrazione.....	51
3.7	Sezione Profilo.....	51
3.7.1	Invio delle modifiche.....	51
3.8	Sezione Allenamenti.....	52
3.8.1	Funzionamento di una ListView.....	52
3.8.2	Elenco degli allenamenti.....	53
3.8.3	Inserimento nuovo allenamento.....	53
3.9	Schede Riepilogative.....	53
3.9.1	Scheda utente.....	53
3.9.2	Scheda allenamento.....	54
3.9.3	Scheda commenti.....	55
3.10	Sezioni di Ricerca.....	55
3.10.1	Ricerca utenti.....	56
3.10.2	Ricerca allenamenti.....	56
3.11	Sezione Live.....	56
3.11.1	Ottenimento della posizione.....	57

3.11.2	Modifiche alla visualizzazione della mappa	58
3.11.3	Rappresentazione degli utenti nella mappa	59
3.12	Gestione delle Notifiche.....	60
3.12.1	Introduzione al servizio	60
3.12.2	Avvio del servizio	60
3.12.3	Funzionamento del servizio.....	61
4	OSSERVAZIONI CONCLUSIVE.....	65
4.1	Osservazioni sui Test Effettuati.....	65
4.1.1	Errori di esecuzione.....	65
4.1.2	Reattività dei servizi.....	65
4.2	Propositi Futuri	66
4.2.1	Risoluzione bug minori	66
4.2.2	Riduzione traffico dati.....	66
4.2.3	Scelta di un dominio appropriato	67
4.2.4	Traduzione in più lingue.....	67
4.2.5	Ottimizzazione grafica	67
4.2.6	Implementazione in altri sistemi operativi.....	67
	BIBLIOGRAFIA	69
	RINGRAZIAMENTI	71

PREFAZIONE

Con questo lavoro ci si pone l'obiettivo di creare un social network che possa aiutare i praticanti del ciclismo, sia a livello agonistico che a livello amatoriale, ad organizzare le loro attività di allenamento, attraverso l'utilizzo di un software sviluppato per smartphone.

Attualmente questo tipo di servizio non è mai stato implementato: l'unica via per poter ottenere un servizio simile è quella di utilizzare pagine su altri tipi di social network oppure effettuare contatti telefonici fra conoscenti.

I dispositivi mobili attuali dispongono di tecnologie per effettuare il collegamento alla rete: la maggior parte dei dispositivi in commercio supporta sia la connessione dati fornita dagli operatori telefonici, sia la connessione tramite Wi-Fi.

Per quanto riguarda i sistemi operativi di riferimento dei dispositivi mobili si presentano diverse alternative, a seconda del dispositivo utilizzato. Le tre principali alternative sono però rappresentate da: iOS, Android e Windows Phone.

Per quanto riguarda iOS attualmente l'ultima versione rilasciata è la versione 7.0. Questo sistema operativo è di proprietà di Apple e viene inserito esclusivamente nei dispositivi rilasciati da questa azienda.

Android è, invece, un sistema operativo open source gestito da Google Inc., che provvede a fornire gli aggiornamenti ed il rilascio delle nuove versioni. L'ultima versione rilasciata è la 4.4, denominata KitKat. Il sistema operativo Android si può trovare su dispositivi di vario genere ma, soprattutto, non è legato ad una specifica casa di produzione di dispositivi mobili. Ognuna di queste case inoltre personalizza la propria versione del sistema operativo, specialmente dal punto di vista grafico. Questa possibilità è data dalla caratteristica open source del sistema operativo accennata in precedenza.

Il terzo sistema citato è Windows Phone. Prodotto dalla Microsoft anche questo sistema operativo può essere ricercato fra dispositivi di case differenti, a differenza di Android comunque questa architettura non è open source. Pur essendo installato su dispositivi di case differenti questo sistema operativo si può trovare principalmente nei dispositivi Nokia di cui la stessa Microsoft ha acquistato il 3 settembre 2013 la divisione Devices & Services.

1. INTRODUZIONE

Prima di procedere alla fase di progettazione del lavoro è bene quindi definire una introduzione degli argomenti che vengono trattati dalla tesi: da una introduzione sullo sport a cui si fa riferimento, alla definizione del social network fino ad un discorso chiarificatore in merito alle applicazioni per smartphone.

1.1 Il Ciclismo ed i suoi Allenamenti

Come prima cosa è bene definire alcuni aspetti che è necessario conoscere relativamente allo sport che si andrà ad analizzare. Il ciclismo è infatti uno sport che può essere praticato in diversi modi e con diversi tipi di attrezzature.

Il ciclismo su strada classico è quello che si può ammirare spesso in televisione, durante eventi come Giro d'Italia e Tour de France tanto per citare le due corse più famose. Salvo casi di percorsi eccezionali questo sport è praticato con biciclette da corsa su strade asfaltate.

Il ciclismo su pista è invece quella specialità che si pratica all'interno di particolari strutture chiamate velodromi. Si tratta di circuiti ovali in cui i ciclisti possono compiere diversi tipi di gare. I velodromi possono essere all'aperto oppure al chiuso ed avere vari tipi di fondi stradali: generalmente si tratta di cemento oppure parquet. In questa categoria di ciclismo vengono utilizzate biciclette da corsa particolari. I mezzi utilizzati sono infatti sprovvisti di freni e cambio marce ma sono dotate di scatto fisso, un particolare sistema che lega la trazione della catena direttamente al movimento della ruota, non consentendo quindi agli atleti di smettere di pedalare improvvisamente.

Si può poi correre il ciclismo su strade sterrate e su percorsi non asfaltati. È questo il caso del ciclocross e del mountain biking. Il ciclocross è praticato su percorsi quasi interamente sterrati e viene praticato utilizzando biciclette da corsa dotate di particolari attrezzature, come ad esempio ruote e copertoni rinforzati. Il mountain biking è invece praticabile sia su strade asfaltate che su strade sterrate e molto sconnesse. In questa specialità vengono utilizzate delle biciclette particolari, chiamate appunto mountain bike, progettate appositamente per praticare questa disciplina in territori boschivi o di aperta campagna.

La specialità di ciclismo di riferimento dell'applicazione è il ciclismo su strada. Generalmente questa specialità è la più praticata anche come attività amatoriale e quella che presenta tracciati più lunghi in termini di chilometraggio.

Le sessioni di allenamento in questo tipo di disciplina possono essere di svariato tipo. Alcuni ciclisti preferiscono praticare il loro sport in tranquillità a ritmi molto blandi per godere il paesaggio, altri preferiscono dare alla loro attività un piglio più grintoso, compiendo allenamenti a velocità più sostenute.

INTRODUZIONE

Un allenamento può inoltre fare riferimento a tipi di tracciato molto differenti: si può svolgere, infatti, un allenamento interamente pianeggiante oppure andare ad affrontare salite di media o elevata difficoltà.

Molti praticanti di ciclismo su strada si organizzano periodicamente in gruppi per effettuare uscite di allenamento, ogni gruppo ha le proprie preferenze di tracciato e di ritmo da sostenere. È però vero che vi sono moltissimi ciclisti che sono costretti a svolgere attività in solitaria, sia per motivi di lavoro, sia perché non riescono a trovare un gruppo di allenamento adatto alle loro esigenze.

L'obiettivo principale dell'applicazione è quello di fondere le due categorie in una categoria sola. Si potranno aggregare fra loro gruppi diversi di una stessa zona, che prima non si conoscevano nemmeno, ma si potrà dare soprattutto la possibilità a ciclisti abituati ad allenarsi in solitaria di inserirsi in gruppi di cui nemmeno si conosceva l'esistenza.

1.2 I Social Network

Lo strumento fondamentale cui si deve fare riferimento quando, al giorno d'oggi, si vogliono far entrare in contatto persone sconosciute è quello del social network. Ma che cos'è un social network?

Un social network non è altro che una particolare piattaforma informatica in cui è possibile interagire con diversi utenti, che condividono le loro informazioni ed i loro interessi.

I tipi di social network possono essere svariati: esistono social network in cui è possibile condividere dati e informazioni dei tipi più svariati. Uno degli esempi più lampanti di questa tipologia di social network è Facebook in cui ogni utente iscritto può condividere praticamente ogni aspetto della sua vita, dalle foto ai video, dai pensieri che gli passano per la testa al suo stato d'animo, e molto altro ancora. Esistono poi social network in cui è possibile condividere video, come ad esempio Youtube, fotografie nel caso di Instagram oppure semplici pensieri come avviene su Twitter.

Per registrarsi ad un social network un utente non deve fare altro che inserire un minimo di dati personali per poter essere riconoscibile ed il gioco è fatto. Quando si parla di profilo di un utente in un social network si fa riferimento appunto ai suoi dati personali che vengono resi pubblici e quindi visibili agli altri utenti.

Oltre a condividere vari tipi di informazioni un social network consente anche di potersi confrontare con altri utenti inserendo commenti in un elemento caricato da se stessi o da chiunque altro. Questa funzionalità viene implementata generalmente da tutti i social network in quanto ne costituisce il vero e proprio cuore. Attraverso i commenti è infatti possibile confrontarsi su varie tematiche e fornire le proprie opinioni.

INTRODUZIONE

Il bello dei social network è appunto quello di rendere tutto il pianeta avvicinabile attraverso la rete. Si possono infatti avviare discussioni e chiacchierate con chiunque utilizzi il servizio, indipendentemente dalla sua provenienza geografica, semplicemente attraverso qualche klik.

Il social network è quindi lo strumento più adatto da implementare nell'applicazione per mettere in contatto ciclisti sconosciuti semplicemente condividendo gli allenamenti che si intende svolgere o prendendo parte ad allenamenti organizzati da altri.

1.3 Le Potenzialità degli Smartphone

È necessario ora fornire una breve introduzione sull'ambiente in cui verrà implementato il social network dedicato al ciclismo.

L'applicazione verrà infatti sviluppata per smartphone, che cosa indica questo termine?

Per smartphone si intende un particolare dispositivo mobile che può fornire diversi tipi di servizi aggiuntivi oltre alle tradizionali funzionalità legate alla telefonia mobile. In uno smartphone è infatti possibile gestire vari tipi di dato come ad esempio la ricezione e l'invio di mail, la gestione di un'agenda elettronica, l'inserimento di promemoria, la navigazione su internet, ecc.

Una delle caratteristiche fondamentali di uno smartphone è però quella di rendere le proprie funzionalità estendibili ad ogni campo. È possibile, infatti, trovare una sorta di negozio digitale in cui gli utenti possono reperire varie applicazioni da installare sul proprio dispositivo. Indipendentemente dal sistema operativo utilizzato la gamma di applicazioni scaricabili è così vasta da rendere praticamente impossibile stilare un elenco di argomenti trattati: si va infatti dalle applicazioni per la gestione di dati bancari fino ai giochi.

Le applicazioni possono essere di due categorie differenti: applicazioni in locale che non necessitano di una connessione ad internet ed applicazioni basate sulla comunicazione di rete, che necessitano di questo tipo di connessione per poter funzionare. I social network devono per forza fare riferimento a questo tipo di architettura.

Uno smartphone può inoltre fungere anche da navigatore satellitare, grazie alle funzionalità di geolocalizzazione che vengono implementate attraverso dispositivi GPS installati nel telefono.

Le caratteristiche elencate rendono quindi l'ambiente degli smartphone perfetto per implementare l'applicazione. La connettività internet rende perfetta la realizzazione di un social network e le funzionalità di geolocalizzazione fanno in modo che sia possibile persino la realizzazione di un servizio in cui i ciclisti possono visualizzare la posizione in cui si trovano altri colleghi su di una mappa.

INTRODUZIONE

Una volta fornite queste precisazioni sull'ambiente in cui si colloca il lavoro da svolgere si può quindi passare alla progettazione vera e propria dell'applicazione e successivamente alla sua realizzazione.

Il lavoro è stato articolato in tre diverse sezioni. Nella prima sezione sarà spiegata tutta la fase di progettazione, nella seconda sarà illustrata la fase implementativa, mentre al termine del lavoro verranno fornite osservazioni e conclusioni sul lavoro svolto.

2 FASE DI PROGETTAZIONE

2.1 Analisi del Domino

La prima domanda da porsi per iniziare la progettazione dell'applicazione è: quali sono i dati da memorizzare?

I requisiti impongono di analizzare due particolari elementi: gli utenti iscritti all'applicazione e gli allenamenti che essi organizzano e svolgono durante la loro attività ciclistica.

2.1.1 Analisi della memorizzazione utente

In questa fase della progettazione è bene ridefinire ancora una volta il concetto di utente che andremo ad analizzare. L'utente in questione è un/una ciclista: i primi dati che andranno inseriti saranno quindi i dati generici di questa persona ovvero il suo nome, il suo cognome e il suo sesso.

Un altro dato che può essere interessante valutare è quello riguardante l'età dell'individuo. A livello informatico è molto più semplice gestire questo dato con la memorizzazione della data di nascita. Se necessario il calcolo dell'età potrà essere eventualmente calcolato in automatico dal sistema. L'applicazione è molto legata alla locazione territoriale di ogni utente, è infatti importante trovare utenti che vivono vicini e possano allenarsi assieme. Sarà molto interessante quindi andare a memorizzare per ognuno di essi la propria residenza, non come luogo di geolocalizzazione (per questo è sufficiente il sistema GPS del dispositivo), ma come vero e proprio luogo, con tanto di dati che si riferiscono alla provincia e alla nazione di appartenenza.

È in fase di valutazione la possibilità di inserire un'immagine del profilo per facilitare il riconoscimento fra gli utenti, questo dato potrebbe infatti risultare utile ma non indispensabile.

Indispensabile è invece l'inserimento di una piccola biografia ciclistica. In questo campo l'utente andrà a inserire a suo piacimento vari dettagli della sua vita sulle due ruote: potrà parlare degli allenamenti seguiti normalmente, dell'attrezzatura in suo possesso, di una sua eventuale carriera agonistica e di quant'altro ritenga interessante agli occhi degli altri ciclisti.

Per finire l'ultimo dato da inserire sarà il campo contenente l'indirizzo mail dell'utente. Questo dato è fondamentale perché, essendo ogni indirizzo mail univoco a livello mondiale, consente di identificare i vari utenti fra loro in caso di omonimia.

2.1.2 Analisi della memorizzazione allenamenti

Il vero obiettivo di quest'applicazione è quello di favorire l'organizzazione di allenamenti di gruppo. Alla luce di questa considerazione si introducono da se i primi due dati relativi ad un allenamento: la data e gli orari di partenza e di arrivo.

Altri dati praticamente obbligatori sono quelli relativi ai luoghi toccati dall'allenamento. Il luogo di partenza è indispensabile, così come quello di arrivo. Può essere molto interessante anche inserire una serie di percorsi intermedi per facilitare la comprensione dell'itinerario.

Altri parametri non indispensabili ai fini organizzativi, ma fondamentali per le decisioni degli utenti, sono quelli riguardanti la lunghezza dell'allenamento: se la lunghezza in chilometri è un dato fondamentale di un itinerario può esserlo anche la durata (indicativa) in minuti della sessione di allenamento.

Un utente poi deve poter scegliere un allenamento anche in base alle proprie caratteristiche, per questo si è ritenuto quasi fondamentale inserire i dati di ritmo e di tipologia, corredati magari da una descrizione testuale contenente vari dettagli aggiuntivi. Per ritmo di allenamento si intende una descrizione della velocità media che si intende tenere durante la sessione, ad esempio si potrà specificare se tenere un ritmo blando, un ritmo allegro o un ritmo forte. La tipologia di percorso invece indica dei dati che si riferiscono al percorso: se in pianura, salita, in collina e così via.

2.1.3 Funzionalità dell'applicazione

Passiamo ora all'analisi delle funzionalità che l'applicazione deve poter garantire ai propri utilizzatori. Occorre anche qui suddividere le analisi in tre sezioni ben distinte: l'analisi di quello che l'utente deve essere in grado di compiere a proposito dei propri dati personali, quello che può fare riguardo alla gestione degli allenamenti e l'analisi di eventuali funzionalità extra.

2.1.3.1 Funzionalità che si riferiscono agli utenti

Innanzitutto occorre fornire gli strumenti di base per garantire l'accesso al servizio, ovvero le sezioni di login e registrazione.

L'utilizzatore deve essere poi in grado di modificare tutti i dati del proprio profilo in ogni momento senza alcuna limitazione. L'unico campo su cui in futuro probabilmente andranno poste delle limitazioni è l'indirizzo mail. Le problematiche relative a questo vincolo possono sorgere in quanto l'attributo in questione serve, come già accennato in precedenza, per identificare univocamente ogni individuo.

Ogni iscritto poi potrà ricercare altri utenti in qualunque momento e visualizzare il profilo di ogni ciclista che utilizza il servizio.

Non si è ritenuto necessario fornire una sorta di blocco utenti per persone indesiderate. Sono due le motivazioni principali che hanno portato a questa scelta.

FASE DI PROGETTAZIONE

Siccome gli allenamenti si svolgono per strada, è davvero impossibile scegliere di evitare a priori persone indesiderate, essendo la strada un luogo pubblico. Osservando poi il comportamento dei vari utenti nei diversi social network il blocco degli amici è parsa spesso una scelta dettata dall'odio, o comunque strettamente collegata ad esso. Siccome uno degli intenti di base dell'applicazione è proprio quello di facilitare il raggruppamento di persone la scelta di inserire un blocco per gli utenti indesiderati è parso in antitesi totale con questo principio.

2.1.3.2 Funzionalità che si riferiscono agli allenamenti

Le operazioni effettuabili sugli allenamenti sono il vero cuore dell'applicazione. Deve essere possibile visualizzare le informazioni di tutti gli allenamenti presenti in memoria.

Ogni ciclista dovrà poi potersi iscrivere a qualunque allenamento. In questo caso va compiuta una breve analisi sulla registrazione degli allenamenti. Se, infatti, un utente dovesse essere iscritto a due allenamenti concomitanti nascerebbe un problema ovvio legato alla questione. Per risolvere questo problema si potrebbe implementare in futuro una funzione di controllo oppure lasciare in mano il problema al buonsenso. Per ora non è necessario comunque soffermarsi ulteriormente sulla questione.

Ogni allenamento deve essere dotato di una sezione relativa ai commenti, in cui sarà possibile svolgere una libera discussione relativa al percorso da svolgere. Ogni utente deve poter commentare qualsiasi allenamento in memoria, sia che vi prenda parte sia che non vi partecipi. È necessaria una precisazione su quest'aspetto: potrebbe sembrare insensato inserire un commento in allenamenti cui non si ha intenzione di partecipare, in realtà l'inserimento di commenti può avere una funzione decisiva proprio per far scegliere se prendere parte o meno alla sessione.

Una volta iscritto a un allenamento un utente deve poter avere la possibilità di ricevere vari tipi di notifiche collegate all'allenamento stesso. Sono stati individuati tre tipi eventi che può essere utile monitorare attraverso l'invio di una notifica: l'inizio imminente di un allenamento a cui si è registrati, l'aggiunta di un commento e la partecipazione di un altro ciclista ad un allenamento personale. Ovviamente sarà data la possibilità di scegliere quali tipi di notifiche si desidera ricevere, potendo cambiare la preferenza in ogni momento.

Siccome è un piacere svolgere un allenamento in gruppo non si è ritenuto necessario limitare il numero di persone che possono essere iscritte a un allenamento.

2.1.3.3 Funzionalità extra

L'applicazione deve fornire la possibilità di visualizzare una mappa in cui siano visibili tutti gli utenti che stanno svolgendo un allenamento nelle vicinanze. In ogni momento un utente potrà scegliere di visualizzare tale mappa per decidere di unirsi ad altri suoi colleghi lungo la strada.

FASE DI PROGETTAZIONE

Per fornire la funzionalità descritta sopra un utente potrà decidere, durante la partecipazione ad un allenamento, inviare la propria posizione in tempo reale al server, in modo da rendersi visibile a tutti gli altri utenti nelle sue vicinanze attraverso la mappa.

2.2 Progettazione delle Entità

Una volta terminato lo studio delle varie parti e delle varie funzionalità che l'applicazione deve gestire si è passati alla progettazione delle entità. In questa fase di lavoro si è scelto di utilizzare il modello E-R, entità-relazioni.

Si è deciso di utilizzare questo modello per vari motivi:

- 1) È molto semplice andare a gestire e rappresentare le varie entità in gioco.
- 2) La rappresentazione delle varie componenti risulta chiara ed immediata anche a chi non ha molta dimestichezza con linguaggi e modelli informatici.
- 3) Siccome l'applicazione nelle fasi successive richiederà la progettazione di un database per immagazzinare i dati questo modello è stato preferito ad altri sistemi di modellazione delle entità, come ad esempio l'UML.

2.2.1 Prima individuazione delle entità

Nella pagina successiva sono mostrate le entità rilevate a seguito della prima analisi del dominio. Non si tratta di una rappresentazione definitiva, ma di una prima rappresentazione schematica delle varie componenti che interagiranno nell'applicazione.

In seguito queste componenti verranno analizzate, corrette e messe in relazione fra loro fino ad arrivare ad un rappresentazione unica e finale del dominio dell'applicazione che dovrà poi essere implementato nella successiva fase di implementazione.

2.2.1.1 Entità Utente

UTENTE
Nome
Cognome
Sesso
Immagine[0-1]
Residenza
Biografia
Data_Nascita
Mail
Password

Nel campo Immagine con [0-1] è rappresentata la cardinalità dell'attributo. Siccome ci si era prefissati di analizzare in seguito la possibile presenza dell'immagine del profilo per gli utenti il campo immagine è stato posto con questa cardinalità per non renderlo obbligatorio al momento dell'inserimento. Tutti gli altri campi, con cardinalità non specificata, sono intesi come unici ed obbligatori.

2.2.1.2 Entità Allenamento

ALLENAMENTO
Luogo_Partenza
Luogo_Arrivo
Lista_Intermedi[0-N] list
Kilometri
Tempo
Tipologia
Descrizione
Ritmo
Data
Ora_Partenza
Ora_Arrivo

FASE DI PROGETTAZIONE

L'attributo `Lista_Intermedi` rappresenta la lista di luoghi che verranno toccati dall'allenamento, ovviamente escludendo i luoghi di partenza e di arrivo. In questo caso la cardinalità [0-N] indica che la lista può avere una dimensione variabile che va da zero ad infinito.

2.2.1.3 *Entità secondarie*

COMMENTO
Testo
Data
Ora

POSIZIONE
Latitudine
Longitudine

L'entità `posizione` dovrà essere legata all'utente per la memorizzazione della sua posizione durante l'allenamento.

2.2.2 **Analisi delle entità individuate**

In questa fase è stata svolta l'analisi delle prime entità rilevate. Il compito di questa fase del lavoro è arrivare a una rappresentazione definitiva delle entità in gioco. Si andranno quindi ad analizzare eventuali dipendenze che possono sorgere fra le diverse entità e fra i loro attributi e si sistemeranno alcuni tipi di dato che potrebbero generare alcuni problemi di rappresentazione.

Di seguito quindi è fornita l'analisi delle entità e dei loro attributi.

2.2.2.1 *Rappresentazione delle date*

Facendo riferimento al tipo `Date` di SQL non ho problemi di generazioni di nuove entità aggiuntive.

Anche nella prima rappresentazione delle entità il tipo di dato utilizzato nel tool di modellazione per le date era `Date`.

Si analizzerà in seguito questo dato durante la fase di implementazione.

2.2.2.2 *Rappresentazione degli orari*

Le possibilità di rappresentazione di quest'attributo che si sono prese inizialmente in considerazione sono due: attraverso una stringa oppure attraverso una nuova entità `Orario`.

Si è però deciso di considerare gli orari non come un'entità, ma come due attributi numerici interi distinti: ore e minuti. Questo evita di perdere tempo per accedere a tabelle di orari che sarebbero inutili e di dover elaborare in maniera inutile delle stringhe.

FASE DI PROGETTAZIONE

Nell'entità allenamento poiché è presente anche l'attributo tempo, che ne misura indicativamente la durata, l'attributo ora di arrivo è ridondante per cui può essere rimosso. Se sarà necessario in fase d'implementazione andare a rappresentare l'orario di arrivo sarà sufficiente sommare all'orario di partenza la durata indicativa dell'allenamento effettuando le opportune conversioni.

2.2.2.3 *Rappresentazione dei luoghi*

Si è deciso di considerare i luoghi come una nuova entità.

Il motivo per cui si è scelto di rappresentare questa entità è legato al fatto che generalmente i percorsi che si seguono durante gli allenamenti toccano spesso certe particolari zone, generalmente delle salite famose. Anche in percorsi diversi, infatti, vengono spessissimo attraversati i soliti luoghi. A trarre un vantaggio da questa rappresentazione sarà sicuramente l'elenco dei luoghi intermedi, che in futuro potrà essere legato da una stretta relazione a questa entità.

I luoghi possono quindi essere individuati da un nome di località, dalla loro provincia di appartenenza e dal loro Stato.

2.2.2.4 *Rappresentazione della residenza di un utente*

Avendo generato l'entità Luogo la si può utilizzare anche per specificare la residenza di un utente.

2.2.2.5 *Ritmo di allenamento*

In precedenza si era parlato di tre esempi di ritmo di allenamento: blando, allegro e forte. Si sono ritenute sufficienti queste tre diverse categorie senza bisogno di inserirne ulteriori. Siccome quindi il numero di ritmi di allenamento è già prefissato credo sia inutile memorizzarli in una tabella che sostanzialmente non subirà alcuna modifica durante il tempo. Per semplicità farò scegliere all'utente il ritmo da un elenco contenente le tre opzioni sopraindicate. Il ritmo selezionato sarà poi inserito come una stringa in memoria.

2.2.2.6 *Tipologia di allenamento*

Anche in questo caso si è seguito lo stesso ragionamento applicato per il ritmo. Le tre tipologie di allenamento individuate sono: pianura, salita e misto. Per misto s'intende un percorso contenete sia pianura che salita circa in egual misura.

2.2.2.7 *Memorizzazione della posizione utente*

In questo caso, invece, si è svolta un'analisi contraria rispetto alle precedenti. Si è giunti alla conclusione che questa entità possa essere rimossa. Per la precisione ho deciso di inglobarla all'interno dell'entità Utente.

Durante la funzionalità live va registrata la posizione istantanea di un utente, che cambierà molto spesso nel tempo durante gli allenamenti. Siccome il riferimento all'utente in questione sarebbe comunque dovuto essere presente all'interno della

FASE DI PROGETTAZIONE

tabella Posizione tanto vale inserire i due attributi latitudine e longitudine direttamente all'interno della tabella Utente. Non è ovviamente possibile, infatti, che un utente possa trovarsi in più luoghi contemporaneamente.

Si è anche svolta una valutazione sull'inserimento di un nuovo attributo booleano per segnalare se l'utente sta usufruendo della funzionalità live, ma non si è ritenuta necessaria in quanto si potrà lavorare nel seguente modo: se latitudine e longitudine sono settate l'utente si sta allenando con la funzionalità attiva, se latitudine e longitudine sono vuote allora l'utente non starà seguendo nessun allenamento.

2.2.2.8 Memorizzazione della durata dell'allenamento

Nell'entità allenamento l'attributo Tempo indica la durata indicativa dell'allenamento in minuti. Può quindi essere rappresentato tranquillamente da un valore intero in quanto si tratta appunto di una stima indicativa e non si cerca una precisione elevatissima nel dato.

2.2.2.9 Immagine di profilo

Si è giunti infine al momento di prendere una decisione sulla memorizzazione o meno dell'immagine del profilo utente. Per evitare problemi di spazio sul database e velocizzare l'esecuzione dell'applicazione ho deciso di eliminare il campo immagine, non ritenendo strettamente necessario l'utilizzo di un'immagine del profilo. In futuro se si disporrà di maggiori risorse si potrà decidere tranquillamente di reinserire questo dato in memoria.

2.2.3 Individuazione delle entità definitive

Si sono quindi modellate le quattro entità, alla luce delle analisi svolte nei punti precedenti. La rappresentazione di queste entità è definitiva e sarà utilizzata per rappresentare il diagramma E-R finale dell'applicazione. Non si è ritenuto necessario fornire i tipi di dato relativi agli attributi in quanto questa precisazione verrà fatta successivamente in fase di implementazione.

FASE DI PROGETTAZIONE

2.2.3.1 Entità Utente

UTENTE
Nome
Cognome
Sesso
Residenza: *LUOGO
Biografia
Data_Nascita
Mail
Password
Latitudine[0-1]
Longitudine[0-1]

Come si evince dallo schema il campo Residenza di un utente è associato all'entità Luogo, che verrà presentata in seguito.

2.2.3.2 Entità Allenamento

ALLENAMENTO
Luogo_Partenza: *LUOGO
Luogo_Arrivo: *LUOGO
Lista_Intermedi[0-N] list: *LUOGO
Kilometri
Tempo
Tipologia
Descrizione
Ritmo
Data
Ora_Partenza_Minuti
Ora_Partenza_Ora

Anche nel caso dell'entità Allenamento con *LUOGO è indicato il riferimento dell'attributo all'entità Luogo.

2.2.3.3 Entità secondarie

COMMENTO
Testo
Data
Orario_Minuti
Orario_Ora

LUOGO
Nome
Provincia
Stato

FASE DI PROGETTAZIONE

Nelle entità secondarie, come si può notare, è stata sostituita l'entità posizione, inglobata all'interno dell'entità Utente, con l'entità Luogo. Quest'ultima entità è l'entità a cui si faceva riferimento nei due punti precedenti relativamente ai luoghi toccati dall'allenamento e alla residenza dell'utente.

2.2.4 Studio delle relazioni fra le entità

Dopo aver rappresentato tutte e quattro le entità che entreranno in gioco nell'applicazione è giunto il momento di studiare le varie relazioni che le legano.

Per relazione fra entità s'intende un legame stretto fra una, due o più entità. Un esempio di relazione può essere: "Persona possiede Automobile", in questo caso la relazione possiede lega fra loro le due entità Persona e Automobile. Per la precisione s'intende che una persona ben precisa in memoria, detta istanza dell'entità Persona, è legata ad una ben precisa istanza dell'entità Automobile. Occorre precisare che una relazione può esistere anche fra istanze della stessa entità, ad esempio: "Persona è parente di Persona".

Una relazione poi è definita anche in termini della sua cardinalità. Un esempio del caso è probabilmente più chiarificatore: "Foresta contiene Albero", la relazione contiene è una relazione uno a molti. Un'istanza di Albero infatti può essere associata ad un'unica istanza di Foresta, mentre un'istanza di questo tipo può ovviamente essere associata a più istanze di Albero, in quanto è chiaro che una foresta per essere definita tale deve essere formata da più alberi. Molto utilizzata è anche la relazione molti a molti, eccone un esempio: "Persona visita Città", una istanza di Persona può aver visitato più di una città e quindi esservi associata; in senso opposto però anche una istanza di Città può essere legata a più istanze di Persona in quanto è evidente che una città può essere visitata da molte persone.

Una volta fornita questa breve introduzione sulle relazioni si andrà ad esaminare tutte quelle che sono presenti fra le quattro entità dell'applicazione.

2.2.4.1 Individuazione delle relazioni principali

Per relazioni principali s'intendono le relazioni presenti fra le due entità principali: Utente e Allenamento. Innanzitutto per essere presente in memoria un allenamento deve essere inserito da un utente: quest'osservazione porta all'inserimento della relazione "Utente genera Allenamento". Ogni utente potrà inserire più allenamenti, ma non sarà nemmeno obbligato ad inserirne qualcuno, ogni allenamento dovrà invece essere, per forza di cose, associato ad uno ed un solo utente.

Una seconda relazione fondamentale da introdurre ai fini del corretto funzionamento dell'applicazione è "Utente partecipa a Allenamento". Un utente può partecipare a più allenamenti, ma può anche risultare non iscritto a nessun allenamento (ad esempio questa situazione si presenterà sempre non appena un utente avrà completato le procedure di registrazione al primo utilizzo dell'applicazione). Un

FASE DI PROGETTAZIONE

allenamento, invece, potrà essere seguito da molte persone. In questo caso però è interessante fare una particolare osservazione: nel caso in cui tutti gli utenti iscritti ad un allenamento decidessero di non prenderne parte per un qualche imprevisto occorre fornire l'opportunità che un allenamento possa essere presente in memoria anche senza nessuna persona iscritta.

Le due relazioni appena presentate sono assolutamente necessarie ai fini di un funzionamento minimo dell'applicazione, senza anche solo una delle due il programma diventerebbe inutilizzabile.

2.2.4.2 Individuazione delle relazioni secondarie

Per relazioni secondarie s'intendono quei legami che non sono strettamente necessarie al funzionamento di base dell'applicazione, ma si rendono necessarie per implementare alcune particolari funzionalità.

Una di queste funzionalità è legata alla possibilità di inserire dei commenti all'interno di un allenamento. Per implementare questo particolare servizio è necessario introdurre due relazioni: "Utente inserisce Commento" e "Allenamento contiene Commento". È fornita di seguito un'analisi di queste due relazioni:

- "Utente inserisce Commento": questa relazione serve per legare ogni commento all'utente che l'ha inserito. Un utente non è chiaramente obbligato ad inserire nessun commento, ma potrà farlo in qualunque momento andandone ad inserire quanti ne desidera (si ritiene sottinteso il buonsenso di ogni utente nell'inserimento di commenti inutili o superflui). Ogni commento invece deve essere associato ad uno ed un solo utente, colui che appunto lo ha scritto.
- "Allenamento contiene Commento": questa relazione esprime, invece, il legame presente fra un commento e l'allenamento cui esso è associato. Ogni commento sarà associato ad uno ed un solo allenamento: non vi è quindi la possibilità di inserire commenti multipli relativi a più allenamenti contemporaneamente. Siccome, come detto al punto precedente, un utente non è obbligato ad inserire nessun allenamento ne segue in maniera ovvia che anche un allenamento potrà essere privo di commenti. Quest'ultima situazione si presenta chiaramente anche quando un allenamento è appena stato creato.

Le altre relazioni secondarie vengono individuate dall'entità Luogo ogni qualvolta essa si presenta come tipo di dato per un campo delle entità Utente e Allenamento.

Nell'entità Utente il campo Residenza è associato a Luogo. Questo legame è rappresentato dalla relazione "Utente vive in Luogo". Ogni utente è tenuto a specificare uno e un solo luogo di residenza. Siccome l'entità Luogo è utilizzata

FASE DI PROGETTAZIONE

anche per esprimere altri tipi di legami non è obbligatorio che ogni luogo sia associato alla residenza di un qualche utente.

Per quanto riguarda, invece, l'entità Allenamento, le associazioni con Luogo sono ben tre: Luogo_Partenza, Luogo_Arrivo e Lista_Intermedi. Ecco la loro analisi:

- **Luogo_Partenza:** per rappresentare questo campo si utilizza la relazione "Allenamento parte da Luogo". Ogni allenamento dovrà aver specificato uno e un solo luogo di partenza, mentre un luogo può essere partenza di più allenamenti differenti. Anche in questo caso non è obbligatorio che un particolare luogo inserito in memoria sia associato ad un luogo di partenza di un qualche allenamento.
- **Luogo_Arrivo:** la relazione introdotta è "Allenamento arriva a Luogo". Per questa relazione valgono tutte le osservazioni fatte per un luogo di partenza.
- **Lista_Intermedi:** questo campo rappresenta invece una lista di più istanze dell'entità Luogo. Si è introdotta quindi una relazione "Allenamento passa da Luogo". Un allenamento può toccare più luoghi intermedi diversi mentre un luogo può essere attraversato anche da più allenamenti (questa era la causa principale che aveva portato all'introduzione di Luogo come entità a se stante). Siccome questo elenco non era stato definito in precedenza come obbligatorio non è necessario fornire per forza almeno un luogo di passaggio. Quest'ultima osservazione porta ovviamente alla conclusione ovvia che una particolare istanza di Luogo non è necessariamente legata a qualche istanza di Allenamento dalla relazione "passa per".

2.2.4.3 Rappresentazione del modello E-R definitivo

Si è giunti a questo punto alla rappresentazione definitiva del modello E-R dell'applicazione. In questo schema è stato inserito anche il concetto di chiave (id) di un'entità. La chiave di un'entità è uno dei suoi campi, scelto per identificare in maniera univoca le varie istanze di quel tipo. Non sarà possibile quindi inserire in memoria due istanze con lo stesso valore nel campo indicato come chiave. Se in un'entità sono indicati come campo chiave due o più attributi si è di fronte ad un caso di chiave multipla. Due o più entità potranno quindi avere gli stessi valori in uno dei campi indicati, ma non potranno avere valori uguali in tutti i campi chiave contemporaneamente.

Nell'elenco sottostante sono indicati tutti i campi scelti come chiave:

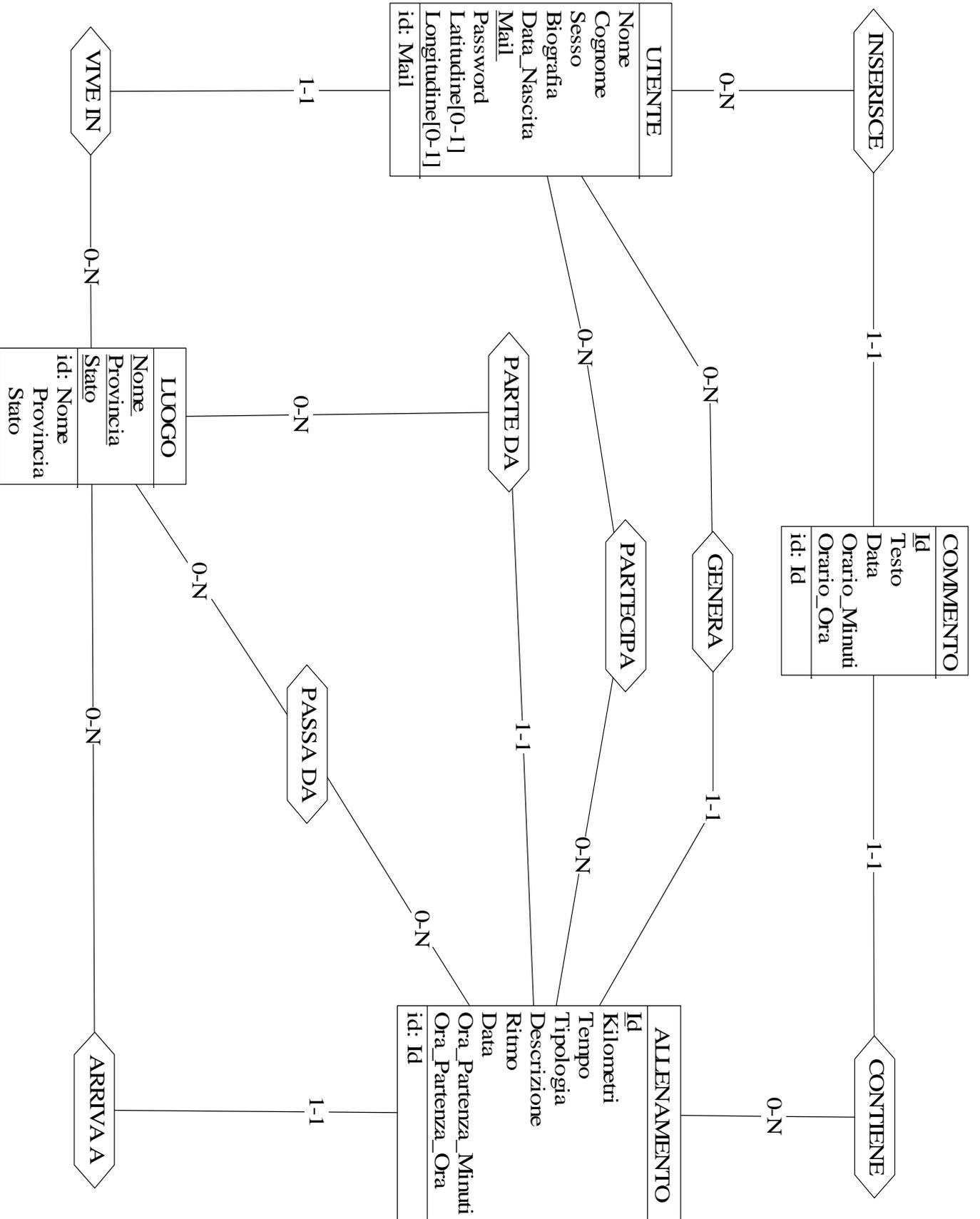
- **Utente:** il campo scelto come chiave è Mail in quanto gli indirizzi e-mail sono univoci a livello planetario. Quest'osservazione riprende un'osservazione fatta in precedenza nell'analisi del dominio.

FASE DI PROGETTAZIONE

- Allenamento: siccome non è stato possibile individuare una chiave composta dai campi di questa entità si è deciso di introdurre un nuovo campo Id, che fungerà da chiave.
- Commento: anche in questo caso è stato necessario introdurre lo specifico campo Id.
- Luogo: in questo caso è stata scelta una chiave multipla formata da tutti i campi dell'entità. Potranno quindi esistere luoghi con lo stesso nome ma in province o stati diversi.

È necessario precisare ulteriormente che la scelta di inserire un nuovo campo denominato Id è stata fatta solo in casi estremi in cui non fosse possibile andare ad identificare univocamente istanze di un'entità attraverso i campi definiti in precedenza.

Nella pagina seguente si può trovare la rappresentazione finale del modello E-R.



2.3 Studio delle Varie Funzionalità Interne

In questa sezione è descritta l'analisi e la progettazione delle varie funzionalità connesse all'applicazione. Queste funzionalità non sono strettamente riferite alle funzionalità dell'applicazione descritte in precedenza, ma a comportamenti interni che richiedevano una analisi prima di passare alla fase di implementazione.

Per la precisione le due funzionalità interne che si andranno a esaminare sono: la ricerca di utenti o allenamenti e la progettazione del server.

2.3.1 Funzionalità di ricerca

La funzionalità di ricerca deve consentire all'utente utilizzatore di poter ricercare altri utenti oppure degli allenamenti. L'obiettivo di questa fase di progettazione è capire cosa può essere necessario specificare nei campi di ricerca.

Si suddividono quindi i due ambiti di ricerca (utenti ed allenamenti) e si analizzano:

- Per quanto riguarda la ricerca degli utenti può essere sensato ricercarli principalmente per nome e cognome. Ritengo utile anche poter ricercare un utente in base alla propria zona di residenza. Per rendere più rapida la ricerca per zona verrà fornita la possibilità di effettuare solamente la ricerca di persone con la stessa provincia e lo stesso Stato di residenza dell'utente. In questo modo l'utilizzatore potrà cercare velocemente altri utenti nelle proprie vicinanze senza dover perdere tempo a scrivere ogni volta la sua zona di residenza, già memorizzata nel database.
- Per quanto riguarda gli allenamenti i campi di ricerca possono essere numerosissimi. La ricerca più sensata è quella ovviamente relativa alla zona di partenza, che comunque sarà da rendere molto simile alla ricerca per zona negli utenti; si darà quindi la possibilità di ricercare allenamenti nelle vicinanze della propria residenza, allenamenti in una zona specifica oppure si potrà direttamente ricercare senza specificare la zona.
Non si è ritenuto indispensabile inserire una ricerca specifica per quanto riguarda la zona di arrivo, sarà l'utente una volta visualizzati gli allenamenti risultanti dalla ricerca a decidere se partecipare o meno a seconda anche della zona di arrivo visualizzata.
Un'altra ricerca molto sensata riguardante un allenamento è ovviamente quella relativa a data e ora di partenza, utilissima se combinata alla ricerca in luoghi limitrofi.
Non si è reputato necessario inserire altri tipi di ricerche sugli allenamenti. Eventuali selezioni per tipologia, ritmo, durata saranno eseguite dall'utente, così come accade per il luogo di arrivo.
Per facilitare la selezione di allenamenti post-ricerca si potrebbe fornire la possibilità di ordinare i risultati secondo alcuni parametri come ritmo, chilometraggio, tipologia e durata.

2.3.2 Funzionalità lato server

Per server si intende quel componente, raggiungibile tramite la rete, in cui verranno memorizzate tutte le informazioni inserite dagli utenti. Il server dovrà fornire quindi tutti gli strumenti necessari per inserire i dati di ogni entità, eliminarli, modificarli oppure semplicemente consultarli.

Si è ritenuto necessario eseguire una progettazione delle funzionalità che dovranno essere implementate dal server cui l'applicazione dovrà collegarsi.

Ogni particolare richiesta che l'applicazione andrà a fare al server dovrà essere implementata in un servizio differente. Questa frammentazione dei servizi consentirà una maggior chiarezza anche in fase d'implementazione.

È inoltre necessario fornire anche un piccolo abbozzo di protocollo di comunicazione che server e applicazione dovranno rispettare.

Per comunicare col server l'applicazione non dovrà fare altro che collegarsi al servizio opportuno e fornire tutti i parametri richiesti.

La risposta del server, invece, deve essere leggermente regolamentata; si è pensato di far inviare dal server la risposta sotto forma di stringa, che potrà assumere una fra le seguenti tre formattazioni:

- Se la risposta è la stringa "1" significa che il server ha portato a buon fine l'operazione richiesta e non ha avuto bisogno di inviare alcun dato aggiuntivo;
- Se la risposta è la stringa "0" significa che vi è stato un errore lato server nell'operazione richiesta, ad esempio un errore di login;
- Se il risultato è una qualsiasi altra stringa significa che il server ha portato a buon fine l'operazione richiesta ed ha inviato un risultato. Il formato di questo risultato andrà deciso in fase d'implementazione.

2.4 Progettazione Estetica

In questa fase viene fornita una breve descrizione delle scelte stilistiche effettuate durante la progettazione dell'applicazione. Questa fase non è assolutamente da sottovalutare poiché una grafica scadente o mal progettata può portare ad una scarsa usabilità dell'applicazione e ad un conseguente abbandono da parte degli utenti. Un'applicazione, infatti, oltre ad essere utile deve anche essere comoda e di facile utilizzo.

Non avendo a disposizione strumenti, capacità e tempo per creare una grafica estremamente accattivante ed elaborata si è deciso di optare per la semplicità. Si è scelto

FASE DI PROGETTAZIONE

di rendere ogni componente grafico il meno confusionario possibile, adottando uno stile minimale.

2.4.1 Scelta dei colori

La prima scelta fondamentale da compiere è legata ai colori di riferimento dell'applicazione.

In questa scelta ci si è collegati a vecchie e gloriose combinazioni di colore utilizzate nel ciclismo di qualche anno fa. Gli appassionati (e forse non solo loro) non possono non aver mai sentito pronunciare la famosissima frase: <<Un uomo solo è al comando; la sua maglia è biancoceleste; il suo nome è Fausto Coppi.>> Si è partiti proprio da quel biancoceleste per effettuare la scelta dei colori.

I colori individuati sono dunque i seguenti:

- Siccome il celeste in questione era probabilmente troppo chiaro è stato modificato facendolo tendere ad un azzurro più vivace.
Il colore scelto è il seguente:  ed ha codice RGB: #1E90FF.
- Per quanto riguarda il bianco si è pensato di adottare una leggera sfumatura di celeste, per ridurre il contrasto fra i due colori.
Il colore scelto è il seguente  ed ha codice RGB: #F0F8FF.
- Unitamente a questi due colori è stato aggiunto un terzo colore per alternare campi differenti all'interno della stessa pagina. Si tratta in realtà solo di una versione del celeste più scura, in questo modo l'utente potrà ad occhio andare a distinguere parti diverse della pagina in maniera molto più intuitiva.
Il colore scelto è il seguente:  ed ha codice RGB: #376FF0.

2.4.2 Presentazione dell'icona

Una volta scelti i colori è stato quindi possibile progettare un'icona per l'applicazione basata su di essi. L'icona individuata è la seguente:



2.4.3 Descrizione dei vari layout

In questa fase è stata eseguita una progettazione esclusivamente grafica dei vari layout dell'applicazione ed è stata progettata la sistemazione delle varie pagine, per rendere l'applicazione il più veloce possibile da utilizzare.

FASE DI PROGETTAZIONE

L'applicazione è stata strutturata in una pagina principale, in cui sarà possibile navigare velocemente all'interno di cinque sezioni, individuate come sezioni fondamentali. Da queste sezioni sarà possibile accedere a ulteriori pagine, che non era possibile inserire all'interno delle sezioni della pagina principale.

Si ricorda che ogni layout visualizzato in anteprima contiene dati a scopo puramente dimostrativo, non ancora collegati al funzionamento stesso dell'applicazione, e fa riferimento ad una implementazione in ambiente Android.

2.4.3.1 Pagina principale

Come già descritto sopra, questa pagina è stata suddivisa in cinque sezioni accessibili tramite un menu a scorrimento situato nella parte alta dello schermo e sempre visibile all'interno della pagina.

È ora fornita la descrizione delle cinque sezioni presenti nella pagina principale:

- **Sezione Profilo:** mostra i dettagli del profilo dell'utente.
Per separare maggiormente i dati e rendere l'applicazione meno caotica si è suddiviso il profilo in tre parti: Dati Personali, Dati Provenienza, Biografia.
Ogni parte sarà selezionabile attraverso un sottomenu.
All'interno dei Dati Personali l'utente avrà la possibilità di modificare i dati relativi a nome, cognome, sesso e data di nascita.
All'interno dei Dati Provenienza l'utente potrà andare a gestire i dati della propria residenza, il riferimento di questi dati è l'entità Luogo.
All'interno di Biografia, invece, l'utente avrà la possibilità di scrivere la propria biografia ciclistica.
Per ogni campo modificabile attraverso questa sezione sarà possibile, tramite un semplice click, andare a modificare in ogni momento questi dati.

The image displays two side-by-side screenshots of the application's profile page. Both screenshots feature a blue header with four menu items: 'PROFILO', 'ALLENAMENTI', 'RICERCA', and 'LIVE'. Below the header, there are three radio buttons for selecting a section: 'Dati Personali', 'Dati Provenienza', and 'Biografia'.
The left screenshot shows the 'Dati Personali' section selected. It contains the following fields: 'Nome: Christian', 'Cognome: Gamberini', 'Nato il: 17/09/1990', and 'Maschio' (selected) / 'Femmina'. Each field has a blue pencil icon to its right, indicating it is editable.
The right screenshot shows the 'Dati Provenienza' section selected. It contains the following fields: 'Luogo: VILLANOVA', 'Provincia: RAVENNA', and 'Stato: ITALIA'. Each field also has a blue pencil icon to its right, indicating it is editable.

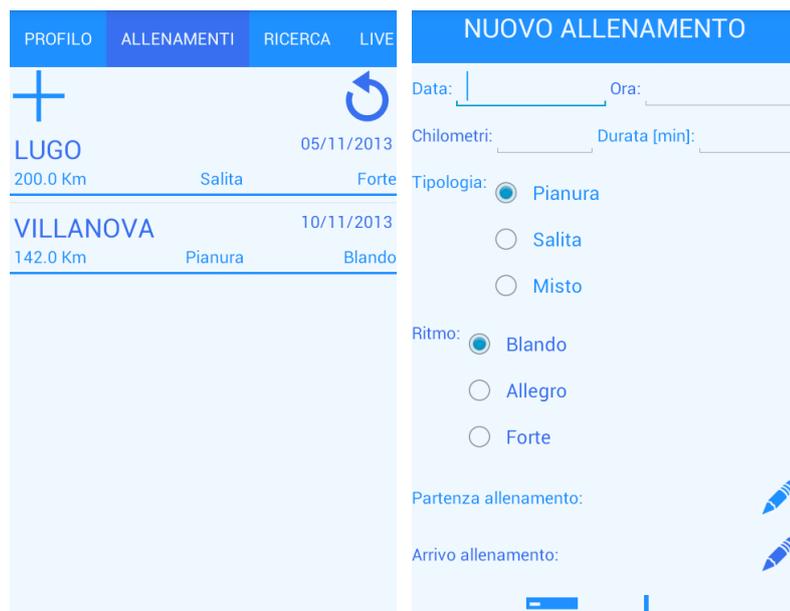
FASE DI PROGETTAZIONE



- Sezione Allenamenti: mostra l'elenco di tutti gli allenamenti a cui l'utente risulta iscritto.

La lista potrà essere aggiornata in qualunque momento.

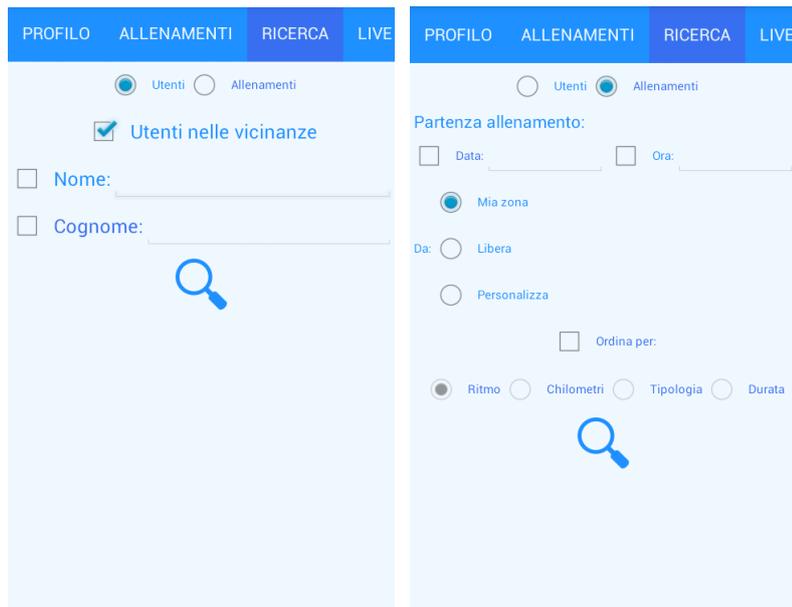
In questa sezione inoltre l'utente potrà andare ad aprire la pagina di creazione allenamento, per organizzare un proprio allenamento, che verrà automaticamente aggiunto all'elenco.



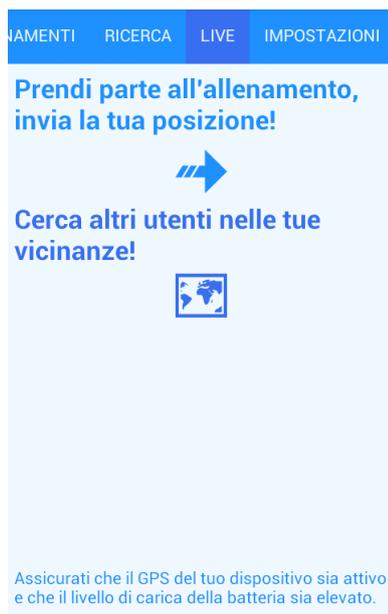
- Sezione Ricerca: in questa sezione sarà possibile effettuare le ricerche, sia di allenamenti che di altri utenti. Come già fatto per la sezione profilo anche in questo caso si è diviso tutto in due parti, accessibili tramite sottomenu: ricerca utenti e ricerca allenamenti.

FASE DI PROGETTAZIONE

Le progettazioni dei layout di ricerca ha ovviamente tenuto conto delle analisi effettuate in precedenza relative a questo argomento.



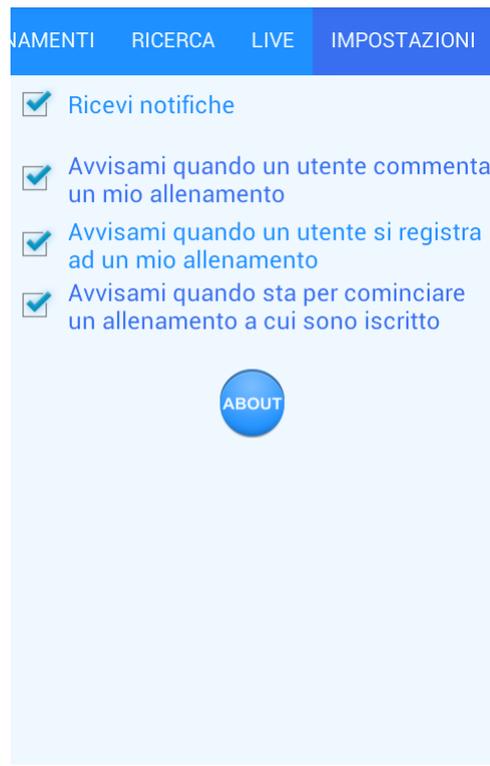
- Sezione Live: qui l'utente potrà iniziare ad inviare la propria posizione agli altri utenti e andare ad aprire la pagina contenente la mappa per visualizzare la loro posizione.



- Sezione Impostazioni: da qui l'utente potrà settare le impostazioni relative alla ricezione di notifiche. Le notifiche potranno essere disabilitate in blocco oppure selezionate singolarmente. Come descritto in precedenza le notifiche gestibili sono: avviso di un commento effettuato su di un proprio allenamento, avviso di una registrazione di partecipazione ad un proprio

FASE DI PROGETTAZIONE

allenamento, avviso di un inizio imminente di un allenamento a cui si è iscritti.



2.4.3.2 *Pagine di introduzione*

Queste pagine sono state create per permettere all'utente di registrarsi e di effettuare il login qualora l'applicazione non sia in grado di effettuarlo automaticamente.

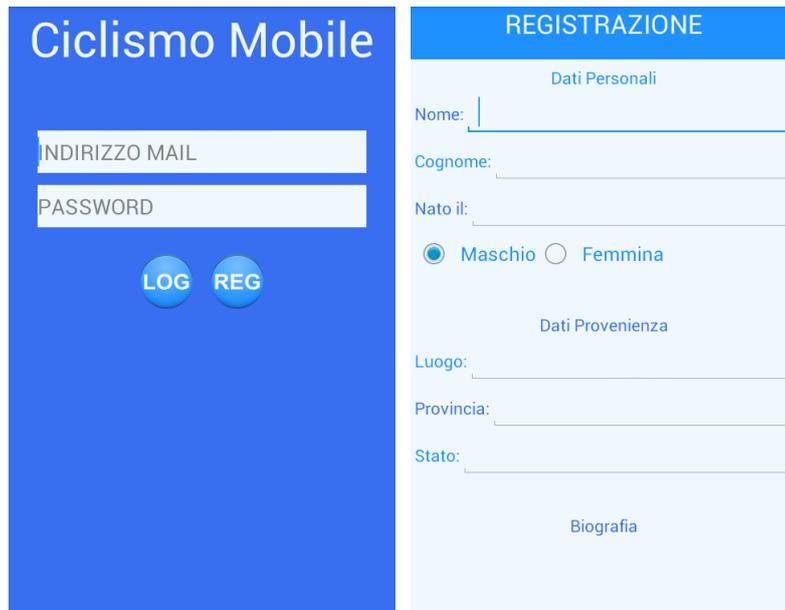
La prima pagina a cui si avrà accesso è quella contenente due campi: uno per la mail utente ed uno, criptato, per inserire la password.

Sotto questi due campi sono disponibili due pulsanti per effettuare login e registrazione.

Premendo login se i dati sono corretti si verrà indirizzati alla pagina principale descritta sopra; premendo registrazione si verrà indirizzati alla pagina di registrazione utente.

Nella pagina di registrazione l'utente andrà ad inserire tutti i dati che potranno poi essere gestiti dalla pagina principale una volta terminata la registrazione.

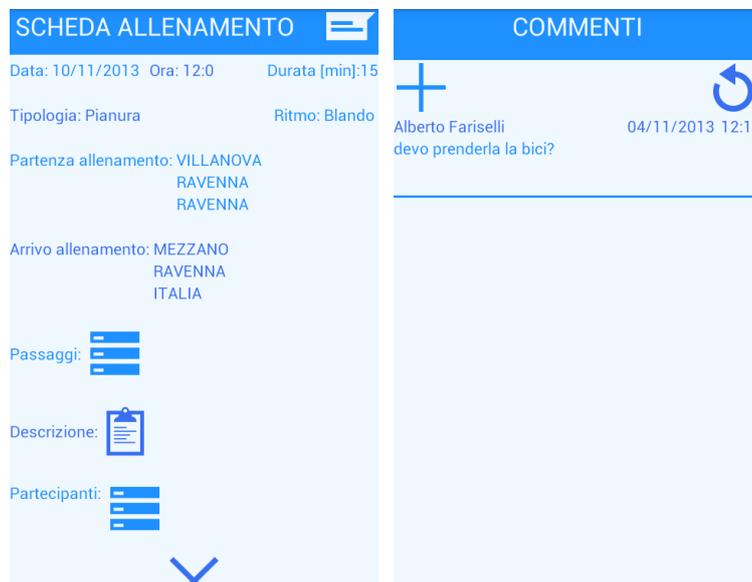
I layout delle pagine di introduzione sono fornite nella pagina seguente.



2.4.3.3 Schede specifiche

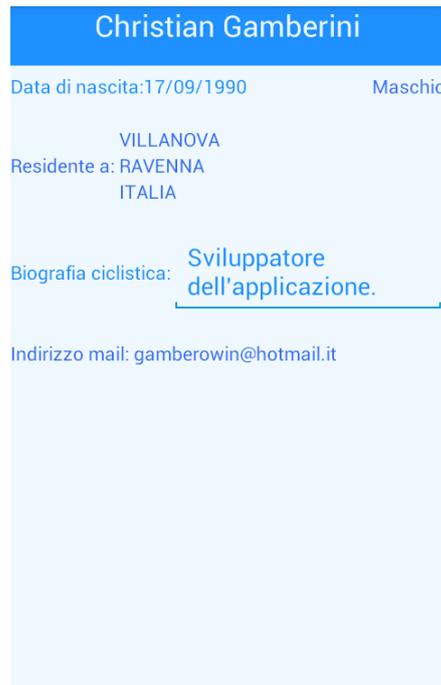
Con schede specifiche si intendono quelle schede che contengono dei dati diversi a seconda di quello che verrà selezionato dall'utente. Queste due pagine sono: scheda allenamento e scheda utente.

La scheda allenamento consente all'utente di visualizzare tutti i dati relativi ad un allenamento, compresi i luoghi di passaggi e l'elenco di utenti iscritti al momento della visualizzazione. Da qui inoltre l'utilizzatore potrà decidere di partecipare o di ritirarsi dall'allenamento visualizzato. In questa pagina poi sarà possibile accedere alla pagina dei commenti, in cui si potranno visualizzare tutti i commenti all'allenamento e se ne potranno aggiungere sempre dei nuovi.



FASE DI PROGETTAZIONE

La scheda utente consente di visualizzare i dati di un utente iscritto all'applicazione. Si potranno visualizzare, oltre al nome ed al cognome, la data di nascita, il sesso, la residenza e la biografia ciclistica.



Queste due schede sono accessibili da diverse pagine, ad esempio dalle pagine di risultati di ricerca o dalla pagina dell'elenco degli allenamenti. La pagina utente è accessibile anche selezionando uno degli utenti partecipanti ad un allenamento.

2.4.3.4 Scheda Live

Questa scheda contiene la mappa in cui sarà possibile monitorare in tempo reale la posizione degli utenti che si trovano nelle vicinanze e che hanno attivato l'invio di dati live dalla relativa sezione nella pagina principale. Proprio dalla sezione principale è possibile aprire la scheda in questione in qualunque momento, previa attivazione del segnale GPS, necessario per visualizzare gli utenti nelle proprie vicinanze.

Il layout di questa sezione è disponibile nella pagina seguente.



2.5 Conclusione Fase di Progettazione

Una volta terminata la fase di progettazione si passa alla fase di implementazione in cui verrà realizzata concretamente l'applicazione.

È necessario specificare che nella prossima fase si potranno apportare modifiche a qualunque componente specificato in questo capitolo. Verranno inoltre analizzate tutte le parti lasciate in sospeso fino ad ora.

3 FASE DI IMPLEMENTAZIONE

La fase del lavoro che si introduce ora riguarda la realizzazione vera e propria dell'applicazione. Per eseguire questa fase è stato scritto del codice in diversi linguaggi. Il codice scritto verrà riportato solo se ritenuto strettamente necessario; verrà piuttosto fornita una descrizione in linguaggio naturale, che è più comprensibile a tutti.

3.1 Conversione delle Tabelle del Modello Relazionale

La prima azione da compiere in fase di implementazione è convertire il modello relazionale generato in fase di progettazione in tabelle, che rappresenteranno la vera e propria organizzazione del database a livello implementativo.

Nel caso in cui in una tabella dovesse comparire un campo chiave di tipo intero (int) questo campo viene dichiarato come auto incrementante, viene cioè generato automaticamente dal database un id crescente, per ogni nuova istanza creata nella tabella.

Siccome si tratta di una rappresentazione molto vicina alla pratica si è deciso di presentare le tabelle qui di seguito piuttosto che in fase di progettazione. I campi evidenziati in grassetto rappresentano gli attributi chiave delle entità.

3.1.1 Tabella Allenamento

I dati relativi a partenza ed arrivo rappresentano le relazioni uno a molti fra Allenamento e Luogo: “Parte da” e “Arriva a”.

CAMPO	TIPO
<i>Id</i>	int
Kilometri	float
Tempo	int
Tipologia	char(50)
Descrizione	char(255)
Ritmo	char(50)
Data	date
Ora_Partenza_Minuti	int

FASE DI IMPLEMENTAZIONE

Ora_Partenza_Ora	int
Partenza_Nome	char(50)
Partenza_Provincia	char(50)
Partenza_Stato	char(50)
Mail	char(50)
Arrivo_Nome	char(50)
Arrivo_Provincia	char(50)
Arrivo_Stato	char(50)

3.1.2 Tabella Utente

I campi relativi alla residenza rappresentano la relazione uno a molti fra Utente e Luogo “Vive in”.

CAMPO	TIPO
Nome	char(50)
Cognome	char(50)
Sesso	tinyint(1)
Biografia	char(255)
Data_Nascita	date
<i>Mail</i>	char(50)
Password	char(50)
Latitudine	doulbe
Longitudine	double
Residenza_Nome	char(50)
Residenza_Provincia	char(50)
Residenza_Stato	char(50)

3.1.3 Tabella Luogo

CAMPO	TIPO
<i>Nome</i>	char(50)
<i>Provincia</i>	char(50)
<i>Stato</i>	char(50)

3.1.4 Tabella Commento

Il campo mail rappresenta la relazione uno a molti “Inserisce” fra Utente e Commento ed indica appunto la mail dell’utente che ha scritto il commento. Il campo CON_Id rappresenta la relazione uno a molti “Contiene” fra Commento e Allenamento, indica l’id dell’allenamento al quale il commento fa riferimento.

CAMPO	TIPO
<i>Id</i>	int
Testo	char(255)
Data	date
Orario_Minuti	int
Orario_Ora	int
Mail	char(50)
CON_Id	int

3.1.5 Tabella Partecipa

Questa tabella rappresenta la relazione molti a molti “Partecipa” fra Utente e Allenamento.

CAMPO	TIPO
<i>Id_Allenamento</i>	int
<i>Mail</i>	char(50)

3.1.6 Tabella Passa_Da

Questa tabella rappresenta la relazione molti a molti “Passa da” fra Allenamento e Luogo, indicante i luoghi da cui transita una istanza di Allenamento.

CAMPO	TIPO
<i>Id_Allenamento</i>	int
<i>Nome</i>	char(50)
<i>Provincia</i>	char(50)
<i>Stato</i>	char(50)

3.2 Scelta del Sistema di Implementazione

Una volta dichiarate le tabelle si inizia la vera e propria fase di implementazione, per prima cosa è quindi necessario scegliere il sistema in cui andrà ad operare.

Il sistema operativo scelto è Android. Questa scelta è avvenuta tenendo conto di alcune caratteristiche fondamentali di questo tipo di architettura, specialmente a livello economico. Non disponendo infatti dei pc appositi per realizzare software per iOS e non avendo disponibili dispositivi con Windows Phone la scelta è ricaduta su Android. Questo sistema offre infatti la possibilità di realizzare software utilizzando un ambiente di sviluppo gratuito chiamato Eclipse. Potendo inoltre reperire alcuni dispositivi Android per eseguire i test la scelta è stata pressoché scontata.

La programmazione in ambiente Eclipse per Android avviene facendo riferimento a due tipi di linguaggio: Java, per realizzare le classi implementanti i servizi, ed XML, per la realizzazione dei layout. Questi due linguaggi, Java in particolare, erano stati ampiamente studiati nel corso della carriera accademica pregressa e quindi sono risultati di immediata facilità d’uso.

3.3 Introduzione agli Strumenti Applicativi

È necessario ora fornire una introduzione generale su alcuni strumenti ed alcune meccaniche che verranno utilizzati dall’applicazione in seguito.

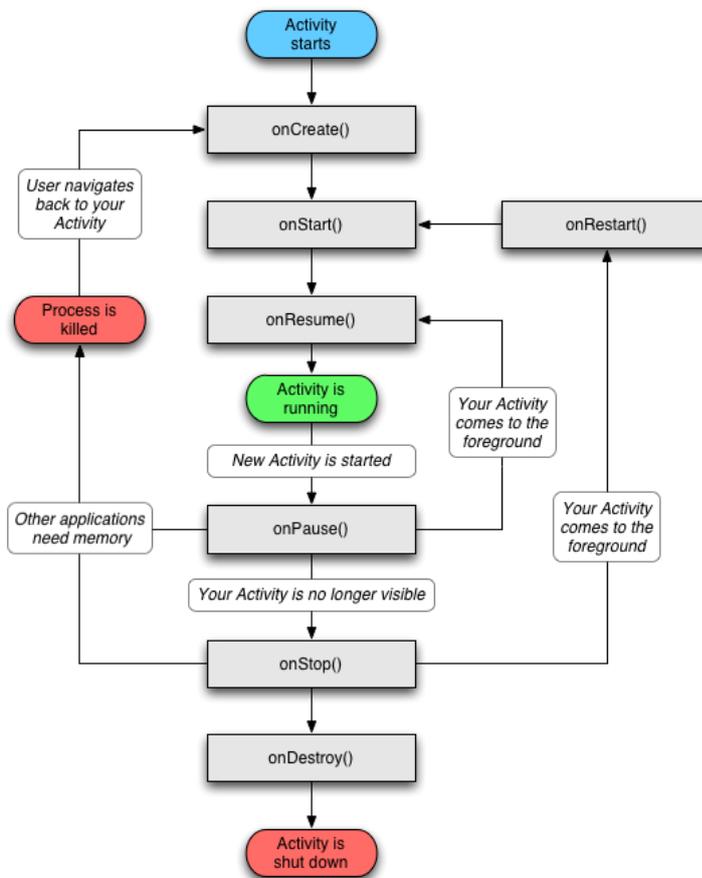
3.3.1 Le activity Android

Per activity in ambiente Android si intendono quei componenti, associati ad un particolare layout grafico, che vengono lanciati e consentono all’utente di eseguire uno specifico tipo di operazione. L’activity principale dell’applicazione è quella che viene avviata non appena viene lanciata l’app. Ogni activity eseguita è associata ad

FASE DI IMPLEMENTAZIONE

un particolare Intent, ovvero una descrizione astratta delle operazione che l'activity deve eseguire.

Ogni activity ha un proprio ciclo di vita che è bene tenere sempre presente in fase di implementazione per evitare errori. Il ciclo di vita di una activity è rappresentato dal seguente schema:



(stackoverflow.com)

3.3.2 Ascoltatori XML

Per implementare le funzioni che si attivano quando l'utente interagisce con l'interfaccia grafica ci si è trovati di fronte a due possibili alternative:

- accedere ai componenti dalla classi Java ed implementare in questo linguaggio i componenti listener, ovvero gli ascoltatori che si attivano quando l'utente o il dispositivo compiono determinate azioni;
- Implementare una funzione Java per ogni ascoltatore ed inserire il nome di tale funzione nel campo onClick dell'elemento XML desiderato nel layout.

Si è optato per seguire sempre la seconda ipotesi in quanto più pulita e veloce.

3.3.3 I Dialog

Con il termine Dialog in ambiente Android si indica una particolare categoria di messaggi popup. Questo tipo di messaggi può essere personalizzato a livello di

FASE DI IMPLEMENTAZIONE

layout dal programmatore a seconda del tipo di messaggio che devono rappresentare. Vi sono Dialog in cui è possibile visualizzare barre di caricamento in attesa che un particolare servizio venga portato a termine, Dialog in cui è possibile inserire dati di ogni tipo, ecc. In ogni Dialog possono essere presenti anche due tipi di pulsanti: un pulsante di conferma ed un pulsante di annullamento, utilizzabili ad esempio nei casi di Dialog in cui viene richiesta la conferma di invio di un qualche dato. Questi pulsanti possono comparire sia singolarmente, sia in coppia, ma possono anche non essere presenti in un Dialog (ad esempio non sono necessari in un messaggio di caricamento). Il programmatore potrà personalizzare appieno le funzionalità collegate a questi pulsanti. Un Dialog può inoltre essere reso cancellabile o meno: nel caso di dialog non cancellabile l'utente non potrà rimuoverlo dallo schermo fino a quando esso non scomparirà in automatico (è sempre questo il caso dei dialog di caricamento).

3.3.4 Shared Preferences

Questo strumento fornito da Android è fondamentale per mantenere dei dati di interesse nella memoria del dispositivo anche quando esso risulta spento, viene riavviato oppure semplicemente quando l'applicazione viene chiusa. È stata creata una classe *PreferenzeCiclismoMobile.java* che implementa le iterazioni con questo servizio di sistema, mantenendo in memoria tutte le etichette con cui il sistema individua i dati dell'applicazione. Tale classe metterà anche a disposizione tutti i metodi per registrare e ottenere i dati presenti nelle preferences.

3.3.5 AsyncTask

Questo strumento serve per far eseguire all'applicazione delle azioni in background attraverso appunto un Task asincrono, ovvero un processo in parallelo. Sarà necessario ricorrere molto spesso a questo tipo di strumento in quanto un'applicazione Android può comunicare attraverso la rete solo all'interno di Task o thread paralleli ma non puoi mai farlo all'interno dei thread principali lanciati dalle activity.

Un AsyncTask è caratterizzato da due sezioni principali:

- La funzione `doInBackground`: rappresenta ciò che il Task andrà ad eseguire in parallelo, questa funzione però non può accedere ai componenti grafici dell'applicazione.
- La funzione `onPostExecute`, che viene eseguita al termine della sezione precedente e può accedere ad ogni componente dell'activity, modificandoli alla luce delle operazioni eseguite in background se necessario.

Si farà riferimento implicito a questo componente ogni qualvolta si menzionerà una comunicazione fra l'applicazione ed il server.

3.4 Implementazione delle classi in Java

Come primo passo di implementazione è stato necessario dichiarare le varie classi Java rappresentanti i dati analizzati nella fase di progettazione. Queste classi consentono al programma di poter gestire i dati in un formato a lui più “digeribile”.

Le classi individuate sono quindi le seguenti: Allenamento, Utente, Luogo, Commento e Tempo. Quest’ultima classe è stata inserita per rappresentare e gestire gli orari.

Vengono riportate di seguito le varie classi ed i relativi attributi. Si sono omesse le parti di codice relative ai metodi di gestione degli attributi: ogni attributo sarà gestito da un metodo di set e da un metodo di get appropriato.

```
public class Allenamento{

    private int id;
    private Date data;
    private Tempo ora;
    private float chilometri;
    private int durata;
    private String tipologia, ritmo;
    private Luogo partenza, arrivo;
    private ArrayList<Luogo> passaggi;
    private String mail;
    private String descrizione;

    //Implementazione metodi
}

public class Utente{

    private String nome,cognome;
    private Date data_nascita;
    private boolean sesso;
    private Luogo provenienza;
    private String biografia;
    private String mail,password;

    //Implementazione metodi
}

public class Luogo{

    private String luogo, provincia, nazione;

    public Luogo(String luogo, String provincia, String nazione){
        this.luogo=luogo.toUpperCase();
        this.provincia=provincia.toUpperCase();
        this.nazione=nazione.toUpperCase();
    }

    //Implementazione metodi
}
```

FASE DI IMPLEMENTAZIONE

```
public class Commento{

    private String testo;
    private int id;
    private Date data;
    private Tempo ora;
    private Utente autore;

    //Implementazione metodi
}

public class Tempo{

    private int ore, minuti;

    //Implementazione metodi
}
```

Queste classi verranno utilizzate in futuro dall'applicazione per gestire tutte le informazioni che verranno inviate o ricevute attraverso le comunicazioni con il server.

3.5 Il Server

Una volta avuta un'idea ben chiara di ciò che si deve realizzare è giunto il momento di scegliere come implementare il server. Si tratta, infatti, di un componente fondamentale dell'applicazione in quanto registra i dati nella rete e permette di ottenerli e di gestirli. Non disponendo di risorse economiche sufficienti per acquistare un dominio internet privato si è deciso di realizzare il server sullo spazio web gratuito Altervista.org, che consente la realizzazione di pagine in linguaggio Php su cui poter implementare tutte le funzioni necessarie per il corretto funzionamento del server. Prevedendo che il numero di utenti sarà inizialmente molto limitato, quantomeno per le fasi di test, questo servizio non ha posto limitazioni considerevoli in termini di traffico e si è rivelato ottimo.

Per ogni servizio sul server si è creata quindi una particolare pagina Php che, una volta presi in ingresso i parametri necessari, effettua tutte le dovute richieste al database e ritorna una stringa implementata utilizzando il semplice protocollo descritto in fase di progettazione. Le richieste al database avvengono tramite query SQL gestite dal linguaggio Php tramite funzioni mySql apposite.

Una volta scelto il sistema di gestione del server si è passati quindi all'implementazione delle tabelle all'interno di esso. Questo processo è avvenuto tramite il tool PhpMyAdmin, fornito gratuitamente dallo spazio web in cui si è deciso di realizzare il server. Le tabelle create sono state generate seguendo appieno la rappresentazione tabellare fornita nei punti precedenti di questo capitolo.

3.5.1 Estensione del protocollo di comunicazione

Come detto in fase di progettazione in questa fase è giunto il momento di specificare il protocollo di invio dei dati da parte del server al client.

Si è deciso di operare come segue: ogni volta in cui sarà necessario inviare un insieme di più attributi, relativi ad una stessa entità, essi verranno separati da un asterisco. Quando, invece, si dovranno inviare dei dati di più entità queste saranno separate dal carattere dollaro ed i loro attributi saranno invece gestiti come descritto in precedenza.

Per chiarire meglio cosa si intende viene fornito di seguito l'esempio di una stringa ritornata dal server, rappresentante un elenco di due utenti:

```
“Christian#Gamberini#1990-09-17#1#LUGO#RAVENNA#ITALIA#Biografia#gamberowin@hotmail.it$Cinzia#Tramonti#1965-02-08#RAVENNA#RAVENNA#ITALIA#Mia madre#cinzia@foo.it”
```

3.5.2 Elenco delle funzioni in linguaggio Php

Si può passare ora ad analizzare i numerosi servizi realizzati sul server in linguaggio Php.

- *“controlloPreRegistrazione.php”*: questa pagina ha come parametri in ingresso la mail di un utente che intende eseguire la registrazione. Ritorna un risultato affermativo nel caso in cui la mail non sia già registrata in database (si ricorda che il campo mail è il campo identificativo della tabella Utente). La funzione è stata implementata per evitare traffico di dati inutili nel caso la registrazione dovesse fallire per la duplicazione dell'id.
- *“registrazione.php”*: questa è la pagina utilizzata per registrare un nuovo utente nel database. Vengono passati alla pagina tutti i dati di registrazione, compresi mail, password e luogo di residenza. Se l'utente viene inserito correttamente la funzione ritorna esito positivo.
- *“login.php”*: questa pagina prende in ingresso una mail ed una password e restituisce esito positivo nel caso fosse presente la combinazione richiesta nelle istanze di Utente. Nel caso in cui la mail non esista oppure non vi sia corrispondenza con la password fornita la pagina ritorna il messaggio di errore.
- *“inserisciCommento.php”*: questa pagina ha come parametri l'id dell'allenamento in cui si dovrà inserire il commento, la mail dell'utente che lo ha scritto ed il relativo testo inserito. Per reperire i dati relativi alla data e all'ora dell'inserimento si è deciso di utilizzare delle apposite funzioni messe a disposizione dal linguaggio Php. In questo modo i commenti potranno

FASE DI IMPLEMENTAZIONE

essere ordinati cronologicamente senza incorrere in errori dovuti a fuso orario o ad una gestione errata delle impostazioni di data e ora del dispositivo inviante la richiesta. Anche in questo caso viene ritornato un messaggio positivo o negativo a seconda dello svolgimento delle operazioni.

- “*inserisciIntermedio.php*”: questa pagina ha come parametri l’id dell’allenamento in questione ed i dati del luogo inserito. Attraverso i dati forniti verrà quindi generata una nuova istanza nella tabella *Passa_Da*. Se il luogo inserito non è ancora presente nella tabella *Luogo* viene generata anche la corrispondente istanza nella tabella *Luogo*. Questo passaggio avviene ogniqualvolta è inserita una istanza di una tabella che contenga dei riferimenti a dei luoghi e non verrà descritto nuovamente in seguito. Viene inviato un messaggio di esito positivo nel caso le operazioni di inserimento vadano a buon fine.
- “*getUtente.php*”: questa pagina ha come unico parametro la mail dell’utente che si vuole visualizzare. Il messaggio di output contiene tutti i dati dell’utente richiesto (ovviamente escludendo la password). La pagina ritorna un messaggio di errore nel caso l’utente non fosse stato rilevato oppure nel caso si presentasse un qualche errore.
- “*modificaDatiUtente.php*”: questa pagina consente di andare a modificare i dati di un utente presente in database. I parametri passati in questo caso possono differire a seconda del tipo di modifica richiesto. Viene passato infatti un parametro indicante il tipo di modifica da apportare (il parametro assume il valore del campo a cui si vuole apportare la modifica, sottoforma di stringa), un parametro indicante il valore da inserire ed il campo contenente la mail dell’utente di riferimento. Nel caso in cui vengano modificati i dati relativi alla residenza sono necessari due ulteriori parametri aggiuntivi per specificare la provincia e la nazione.
- “*ottieniCommenti.php*”: questa pagina consente di ottenere l’elenco dei commenti relativi ad un allenamento il cui id viene passato come unico parametro. In caso di successo vengono inviati per ogni commento i suoi dati ed i dati relativi all’utente che lo ha scritto.
- “*ottieniIscrizioni.php*”: questa pagina fornisce, in caso di esito positivo, l’elenco degli allenamenti a cui risulta essere iscritto l’utente specificato dal parametro mail.
- “*ottieniLuoghiIntermedi.php*”: questa pagina ritorna l’elenco dei luoghi da cui transita un allenamento, specificato dall’unico parametro in ingresso. Si è

FASE DI IMPLEMENTAZIONE

deciso di ritornare questi luoghi separatamente dall'allenamento per non dover stravolgere il protocollo di comunicazione.

- “*ottieniPartecipanti.php*”: questa pagina prende in ingresso l'id di uno specifico allenamento e ne ritorna l'elenco dei partecipanti. Per ogni utente vengono ritornati tutti i dati pubblici.
- “*inviaCoordinate.php*”: questa pagina consente all'utilizzatore, indicato dal parametro mail, di inviare le proprie coordinate geografiche al server, passate come parametri.
- “*ottieniVisualizzabili.php*”: questa pagina ritorna il nome, il cognome e le coordinate geografiche di tutti gli utenti che non le hanno impostate a NULL. Fornisce quindi l'elenco di tutti gli utenti che stanno utilizzando la funzione live. Si è deciso di inviare solamente nome, cognome e coordinate per risparmiare dati inviati, altri dati sono stati considerati inutili ai fini della visualizzazione live.
- “*partecipa.php*”: questa pagina prende in ingresso un id di un allenamento e la mail di un utente ed inserisce l'utente specificato come partecipante all'allenamento. Viene quindi creata una nuova istanza nella tabella Partecipa.
- “*pulisciCoordinate.php*”: prendendo in ingresso la mail di un utente questa pagina provvede a settare le sue coordinate geografiche al valore NULL. Questa funzione verrà richiesta quando l'utente in questione richiederà lo stop della funzionalità live.
- “*creaAllenamento.php*”: questa pagina ha come parametri in ingresso tutti i dati relativi ad un allenamento e provvede alla loro registrazione in database. Viene inoltre registrato l'utente creatore (la mail è passata come parametro) come partecipante nell'apposita tabella. La funzione ritorna un messaggio positivo in caso di avvenuta registrazione.
- “*ricercaAllenamento.php*”: attraverso questa pagina è possibile effettuare la ricerca di un allenamento presente in database. Il primo parametro controllato dalla pagina è relativo alla modalità di ricerca, ovvero ai parametri di cui la ricerca terrà conto per filtrare i risultati. Sono stati individuati sette diversi tipi di modalità con indice di identificazione crescente da zero a sei:
 0. Questa modalità ricerca un allenamento in base al luogo di partenza. Sono sempre da specificare la provincia e la nazione come parametri. Se il luogo non dovesse essere specificato si ricercheranno tutti gli

FASE DI IMPLEMENTAZIONE

allenamenti con partenza nella provincia e nazione indicati. Quest'ultimo tipo di ricerca implementa la ricerca di allenamenti nelle vicinanze di una determinata zona.

1. In questa modalità viene passato un unico parametro di ricerca, rappresentante la data di svolgimento dell'allenamento.
2. Questa modalità riceve in ingresso due parametri identificanti l'ora di partenza dell'allenamento: un parametro specifica l'ora ed il successivo specifica i minuti.
3. Si combinano fra loro le modalità di tipologia 0 e di tipologia 1, ovvero ricerca un allenamento in base alla data ed al luogo di partenza.
4. Questa modalità combina assieme le modalità di tipologia 0 e di tipologia 2.
5. Grazie a questa modalità si combinano le modalità di tipo 1 e 2.
6. Questa è la modalità di ricerca più specifica, combina infatti tutte le modalità di tipo 0, 1 e 2.

Viene poi passato un parametro booleano che rappresenta la richiesta di ordinamento dei risultati. Nel caso in cui l'ordinamento fosse richiesto viene passato un ulteriore valore intero che rappresenta il tipo di ordinamento da effettuare:

1. Questo valore rappresenta l'ordinamento per ritmo di allenamento.
2. Con questo valore si ordinano gli allenamenti in base alla loro lunghezza in chilometri.
3. Attraverso questo valore si ordinano gli allenamenti in base alla loro tipologia, equivale cioè a raggrupparli per tipo.
4. Si ordinano in questo caso gli allenamenti in base alla loro durata in minuti.

Come risultato di ogni ricerca con esito positivo viene ritornato l'elenco degli allenamenti ottenuti. Per ogni allenamento vengono ritornati tutti i dati, escluso l'elenco dei luoghi intermedi.

- *“ricercaUtente.php”*: si tratta della seconda pagina di ricerca, implementante la ricerca di un utente. Come accade nella pagina di ricerca degli allenamenti anche qui si invia una modalità di ricerca ed a seconda della scelta fatta si inviano parametri differenti.

Di seguito vengono elencate le sette modalità di ricerca per gli utenti:

0. Questa modalità ricerca gli utenti provenienti da una determinata provincia e nazione. Implementa la ricerca per zone di residenza descritta in fase di progettazione.

FASE DI IMPLEMENTAZIONE

1. Viene qui ricercato un utente solamente in base al proprio nome di battesimo.
2. Nella modalità numero due si ricercano tutti gli utenti in database che hanno il cognome specificato.
3. Questa modalità combina assieme le modalità 0 e 1.
4. In questo tipo di ricerca vengono sfruttate insieme le modalità 0 e 2.
5. Questa modalità consente di ricercare un utente in base a nome e cognome, si combinano cioè le modalità 1 e 2.
6. Come per la ricerca allenamenti anche in questo caso l'ultima modalità raggruppa insieme le modalità 0,1 e 2.

Nella ricerca degli utenti non si è ritenuto necessario fornire alcun tipo di ordinamento.

Se la ricerca ha avuto esito positivo la pagina ritorna l'elenco degli utente, indicandone tutti i dati pubblicabili.

- “*ritirati.php*”: questa pagina consente ad un utente, specificato dal parametro mail, di ritirarsi da un allenamento a cui risulta iscritto. L'id dell'allenamento viene passato come parametro. La funzione non fa altro che rimuovere un'istanza dalla tabella Partecipa.

3.5.3 Classe server in Java

Per fare in modo che l'applicazione possa comunicare con il server in maniera semplice viene realizzata la classe *ServerCiclismoMobile.java* in cui è stata implementata una funzione per ogni servizio implementato sul server. Per questo ultimo motivo non si ritiene necessario fornire un ulteriore elenco degli stessi servizi, anche se implementati in un linguaggio differente.

Ogni funzione Java prende in ingresso gli stessi parametri richiesti dal servizio Php e genera una stringa rappresentante l'url di riferimento del servizio. I parametri relativi alle stringhe vengono gestiti attraverso la funzione privata *eliminaSpazi(String)* che prende in ingresso una stringa e la rielabora per renderla compatibile con il formato url, andando cioè a sostituire tutti i caratteri invalidi in tale formato con i loro caratteri di escape (ad esempio il carattere spazio verrà sostituito da “%20”). Per quanto riguarda invece la gestione delle date e degli orari è stata implementata la classe *GestoreDateTempo.java* contenente delle funzioni statiche in grado di convertire una stringa rappresentante una data o un'ora valida in un formato compatibile al server (aaaa-mm-gg) e viceversa (viene convertita una data in formato server nel formato gg/mm/aaaa). Per quanto riguarda gli orari vengono gestiti ovviamente tramite la classe *Tempo* dichiarata in precedenza.

Nel caso in cui si presentino degli errori nei parametri inseriti la funzione ritornerà

FASE DI IMPLEMENTAZIONE

esito negativo senza nemmeno effettuare il collegamento col server, in questo modo ci si risparmia di inviare nella rete dati inutili che verrebbero rifiutati.

Una volta generata la stringa url, nel caso in cui non si siano presentati degli errori, ci si collega all'indirizzo che essa rappresenta e si ottiene il risultato sotto forma di stringa. Nel caso in cui il server dovesse restituire un messaggio di errore la funzione ritorna un valore apposito rappresentante uno dei seguenti valori: false per le funzioni booleane, NULL per le funzioni che richiedono un singolo oggetto, un elenco vuoto per tutte le funzioni che devono ritornare un elenco.

Nel caso in cui il risultato sia valido la stringa viene interpretata accuratamente e convertita in un risultato in linguaggio Java. Ad esempio per creare un ArrayList dei risultati di un allenamento si è proceduto nel seguente modo:

1. Attraverso uno StringTokenizer vengono separati i vari allenamenti ottenuti, utilizzando come carattere di separazione il dollaro.
2. Per ogni stringa fornita al punto precedente il costruttore della classe Allenamento dichiara un nuovo StringTokenizer che separa gli attributi (il carattere di separazione è l'asterisco) e li converte nel tipo corretto.
3. L'allenamento creato viene aggiunto all'ArrayList.

Se si dovessero presentare errori di ogni tipo durante la conversione del risultato la funzione ritornerà errore come descritto in precedenza.

Creando la classe proxy del Server si rende mascherato il collegamento web e la creazione dell'url durante la programmazione. Per potersi collegare al server basterà quindi dichiarare una istanza della classe *ServerCiclismoMobile.java* all'interno di un AsyncTask e richiamare la funzione corrispondente al servizio a cui si vuole accedere.

3.6 Procedure di Login e Registrazione

Da questa fase si inizia l'analisi vera e propria di come sono stati implementate le varie componenti e le varie fasi dell'applicazione.

In questa prima sezione viene descritta l'implementazione delle procedure di login e di registrazione dell'utente utilizzatore.

3.6.1 Login automatico

All'avvio dell'applicazione verrà effettuato un tentativo di login automatico. Per implementare questa operazione è stata utilizzata la classe di interfacciamento con le Shared Preferences descritta precedentemente.

Quando l'activity principale viene creata viene mostrato un layout di caricamento ed in background viene lanciato un AsyncTask che andrà ad effettuare il login

FASE DI IMPLEMENTAZIONE

(attraverso il servizio *login.php*) recuperando la mail e la password dalle preferenze. In caso di esito positivo il layout verrà rimpiazzato dalla sezione principale dell'applicazione. I componenti di questa sezione verranno riempiti da un altro task parallelo che richiederà al server i dati dell'utente specificato dalla mail registrata sul dispositivo. In caso di esito negativo invece verrà caricato il layout di login da cui si potrà tentare di effettuare un login manuale oppure di registrarsi.

3.6.2 Login manuale

Una volta inseriti i dati nel campo mail e nel campo password l'utente potrà tentare di effettuare il login premendo l'apposito pulsante (LOG). In caso di successo verranno eseguite le operazioni descritte nel login automatico ed i due campi inseriti verranno salvati nelle preferenze per gli utilizzi successivi. In caso di errore sarà invece visualizzato un dialog (ovvero un messaggio su schermo) di errore e si resterà all'interno di questo layout.

3.6.3 Registrazione

Se si tratta del primo utilizzo dell'applicazione l'utente potrà effettuare la registrazione dallo stesso layout in cui viene effettuato il login manuale. L'utente dovrà inserire la propria mail e la propria password e premere il pulsante di registrazione (REG). Un primo task controllerà sul server che i due dati inseriti siano corretti e che l'utente non risulti già registrato attraverso il servizio *controlloPreRegistrazione.php*. Nel caso in cui quest'ultimo servizio dia esito positivo viene lanciata l'activity di registrazione in cui l'utente potrà inserire i propri dati ed inviarli al server. Nel caso di dati incorretti verrà invece mostrato un dialog di errore. Successivamente in caso di registrazione avvenuta la mail e la password verranno registrati nelle preferenze, l'activity di registrazione verrà chiusa e si verrà reindirizzati all'activity principale che provvederà ad eseguire la procedura di login automatico.

3.7 Sezione Profilo

In questo punto verrà descritta la sezione del profilo, ovvero la prima sezione della pagina principale, accessibile dal menù a scorrimento.

Il caricamento dei dati in questa sezione avviene, come già spiegato in precedenza, in automatico al termine della procedura di login, attraverso il servizio *getUtente.php*.

3.7.1 Invio delle modifiche

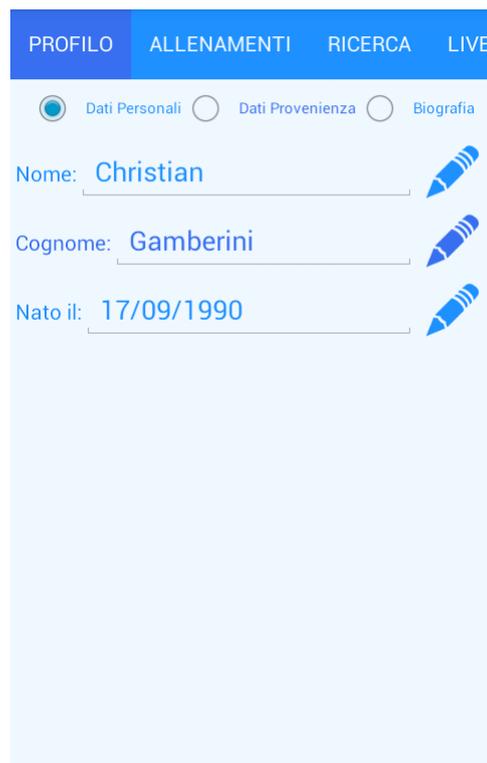
Come già descritto in fase di progettazione l'utente potrà inviare le proprie modifiche andando semplicemente a modificare i dati in questa sezione della pagina e premendo il pulsante a forma di matita a fianco ad essi. Una volta premuto il pulsante l'applicazione invierà al server la modifica del solo attributo a cui il pulsante fa riferimento. Prima di ciò verrà comunque richiesta all'utente una

FASE DI IMPLEMENTAZIONE

conferma dell'operazione tramite la comparsa di apposito Dialog contenente i due pulsanti di conferma e di annullamento.

In questa fase si è apportata una modifica al layout in quanto non si è ritenuto necessario andare a modificare il sesso dell'individuo, che dovrebbe rimanere invariato durante gli utilizzi. Per mantenere un'indicazione sul sesso in questa sezione basterà guardare se a fianco a data di nascita è presente la stringa "Nato il:" o la stringa "Nata il:".

Il nuovo layout della sezione profilo, relativamente ai dati dell'utente è quindi il seguente:



3.8 Sezione Allenamenti

Questa sezione contiene l'elenco degli allenamenti a cui l'utente risulta iscritto. Nell'elenco possono essere presenti sia allenamenti creati dall'utilizzatore che allenamenti creati da altri utenti.

3.8.1 Funzionamento di una ListView

Prima di entrare nello specifico di questa sezione occorre spiegare il funzionamento di un particolare componente grafico di Android: le ListView, che verranno utilizzate spesso anche in seguito.

Una ListView non è altro che un elenco verticale che serve per rappresentare un elenco di dati tutti dello stesso tipo. Uno dei prego fondamentali delle ListView è

che il layout utilizzato per rappresentare una riga può essere completamente e facilmente personalizzato dal programmatore. Ogni riga sarà formattata secondo lo stesso layout ma conterrà quindi dati diversi. Quando poi una riga viene selezionata tramite un click la ListView fornisce l'opportunità di sapere l'indice della riga selezionata, e di conseguenza l'indice dell'elemento nella struttura dati rappresentata. Questo componente verrà utilizzato nell'applicazione ogni qualvolta vi sarà bisogno di rappresentare un qualunque tipo di elenco.

3.8.2 Elenco degli allenamenti

L'elenco degli allenamenti a cui l'utente risulta iscritto viene appunto rappresentato in una ListView in cui sono visualizzati, per ogni allenamento: il luogo e la data di partenza, la lunghezza in chilometri del percorso, il ritmo e la tipologia. L'elenco è aggiornabile tramite l'apposito pulsante ma non viene effettuato un aggiornamento automatico per non appesantire l'esecuzione. Selezionando una riga di questo elenco verrà aperta la scheda dell'allenamento corrispondente, questa vista verrà presentata in seguito.

3.8.3 Inserimento nuovo allenamento

In questa sezione è presente inoltre un pulsante a forma di più. Premendo il pulsante l'utilizzatore potrà andare ad organizzare sotto tutti gli aspetti un nuovo allenamento.

Alla pressione del pulsante verrà avviata una nuova activity di creazione allenamento in cui l'utente potrà registrare tutti i dati della sessione. Per quanto riguarda l'elenco dei luoghi intermedi è possibile visualizzarli attraverso un dialog contenente una ListView. L'inserimento del luogo di partenza, del luogo di arrivo o di un luogo di passaggio avviene attraverso un dialog che li registra in un elenco.

Alla pressione del pulsante di invio dati, in caso non venga visualizzato il dialog di errore, l'applicazione invia al server prima i dati dell'allenamento tramite il servizio *creaAllenamento.php*, successivamente tramite il servizio *inserisciIntermedio.php* vengono inseriti ad uno ad uno tutti i passaggi intermedi dell'allenamento. Una volta inseriti i dati di un allenamento non sono più modificabili in alcun modo per evitare incomprensioni con gli utenti iscritti.

3.9 Schede Riepilogative

Con schede riepilogative si intendo delle particolari activity che mostrano nel loro layout il riepilogo di informazioni di alcune istanze in database di cui si desidera visualizzare le informazioni. I tre tipi di schede visualizzabili sono: scheda utente, scheda allenamento e scheda dei commenti.

3.9.1 Scheda utente

In questa scheda, il cui layout è già stato presentato in fase di progettazione, vengono visualizzate tutte le informazioni relative ad un particolare utente presente in

FASE DI IMPLEMENTAZIONE

database. L'unico dato a non essere ovviamente mostrato è quello relativo alla password.

Per reperire i dati dell'utente l'activity li riceve come extra dall'activity da cui si è scelto di aprire la scheda. In Android è infatti possibile gestire la comunicazione fra activity passando all'intent di riferimento degli attributi o degli oggetti associati ad una etichetta. L'activity aperta potrà poi reperire dal suo intent i dati passati semplicemente facendo riferimento a quell'etichetta.

Nel caso della scheda utente viene passato un oggetto utente come serializzabile. Si è dovuta quindi estendere la classe Utente facendo in modo che implementasse l'interfaccia Serializable. Una volta ottenuto un oggetto serializzabile nella scheda utente basta eseguire un cast ad utente e si avrà la variabile di tipo Utente di cui si vogliono visualizzare le informazioni.

3.9.2 Scheda allenamento

In questa scheda vengono invece visualizzate tutte le informazioni relative ad uno specifico allenamento. Anche in questo caso l'oggetto allenamento viene passato come oggetto serializzabile ed è stato necessario modificare la classe Allenamento come lo si era fatto in precedenza per la classe Utente.

3.9.2.1 Caricamento passaggi intermedi

Siccome rappresentare l'elenco dei passaggi, che può anche essere molto lungo, poteva rendere molto illeggibile la scheda si è deciso di inserirli all'interno di una ListView visualizzabile all'interno di un dialog che si apre ogni qualvolta l'utente preme il pulsante di visualizzazione dei luoghi intermedi. In questa ListView non è stato implementato l'onClik per selezionare le righe.

3.9.2.2 Caricamento dei partecipanti

Per caricare l'elenco dei partecipanti si è seguito lo stesso ragionamento fatto per i passaggi intermedi, con l'unica differenza relativamente all'implementazione dell'onClik nelle righe della ListView. Selezionando, infatti, una riga l'utilizzatore può visualizzare la scheda utente dell'utente selezionato.

Il caricamento dei partecipanti non viene eseguito inoltre all'avvio activity, ma avviene ogni qualvolta l'utente preme il pulsante di visualizzazione partecipanti. Si è scelto di implementare il caricamento in questo modo in quanto l'elenco dei partecipanti potrebbe aggiornarsi da un momento all'altro. Il difetto di questa scelta implementativa è ovviamente quello di rendere più lento il caricamento dei partecipanti, dato che l'applicazione deve richiederli ogni volta al server.

3.9.2.3 Partecipare/ritirarsi da un allenamento

In fondo alla scheda allenamento, in posizione centrale, è presente un pulsante, che l'utente potrà visualizzare in due forme diverse. Se viene visualizzata una puntina significa che l'utente non risulta partecipante all'allenamento, se viene visualizzata

FASE DI IMPLEMENTAZIONE

una croce significa che l'utente è registrato ai partecipanti. Premendo tale pulsante l'utente può quindi decidere in ogni momento quando iscriversi o ritirarsi dall'allenamento.

In realtà i pulsanti implementati sono due e vengono resi visibili o invisibili a seconda dell'iscrizione o meno dell'utilizzatore all'allenamento selezionato. Il pulsante di partecipazione fa riferimento al servizio *partecipa.php*, mentre il pulsante per ritirarsi fa riferimento al servizio *ritirati.php*.

Il dato booleano relativo alla partecipazione dell'utente viene passato come extra all'intent dell'activity.

3.9.3 Scheda commenti

Questa scheda viene aperta premendo il pulsante a forma di fumetto presente nell'intestazione della scheda allenamento. Viene passato come extra all'intent l'id dell'allenamento e la nuova activity appena creata provvede a caricare tutti i commenti relativi a quell'allenamento durante la sua creazione (tramite il servizio *ottieniCommenti.php*). I dati vengono inseriti in un'apposita ListView in cui è possibile, selezionando la riga di un commento, andare ad aprire la scheda dell'utente che l'ha inserito.

In questa scheda sono inoltre presenti due pulsanti: un pulsante a forma di più che fornisce la possibilità di inserire un nuovo commento ed un pulsante a forma di freccia circolare per aggiornare l'elenco. L'inserimento avviene tramite un dialog contenente un campo di testo. Premendo il pulsante di conferma nel dialog il commento viene inserito mediante il servizio *inserisciCommento.php*, la ListView viene poi aggiornata automaticamente in modo da mostrare il nuovo commento inserito.

La ListView può anche essere comunque aggiornata sempre premendo il pulsante a forma di freccia circolare. Come in tutte le ListView aggiornabili presenti nell'applicazione non avviene mai un aggiornamento automatico timerizzato, ma è fornita solamente la possibilità di effettuare l'aggiornamento manuale.

3.10 Sezioni di Ricerca

In questa sezione è possibile effettuare sia la ricerca utente sia la ricerca di allenamenti.

Come già visto in fase di progettazione dei layout ogni parametro di ricerca è stato associato ad una checkBox che permette di individuare la modalità di ricerca.

Per fare in modo che sia possibile effettuare la ricerca di allenamenti e utenti che si trovano nelle vicinanze dell'utilizzatore si è fatto riferimento alle preferenze, in cui viene inserita in fase di registrazione o di modifica la residenza dell'utilizzatore.

FASE DI IMPLEMENTAZIONE

La ricerca è comunque delegata alle activity di ricerca, che vengono avviate alla pressione del pulsante apposito.

3.10.1 Ricerca utenti

Vengono passati a questa activity come extra quattro parametri: il nome, il cognome, la provincia e la nazione. Nel caso in cui non fosse necessario effettuare la ricerca secondo uno dei parametri indicati (checkBox non selezionata) esso viene comunque passato con valore NULL.

Il task di ricerca viene avviato alla creazione dell'activity e fornisce tutti e quattro i parametri alla funzione della classe *ServerCiclismoMobile.java*, che provvede a riconoscere la modalità di ricerca ed a generare l'url corretto, che fa riferimento al servizio *ricercaAllenamento.php*.

3.10.2 Ricerca allenamenti

La ricerca degli allenamenti avviene allo stesso modo della ricerca utenti. In questo caso i parametri passati all'activity sono: data, ora, luogo, provincia, nazione, ordinamento, e tipo di ordinamento.

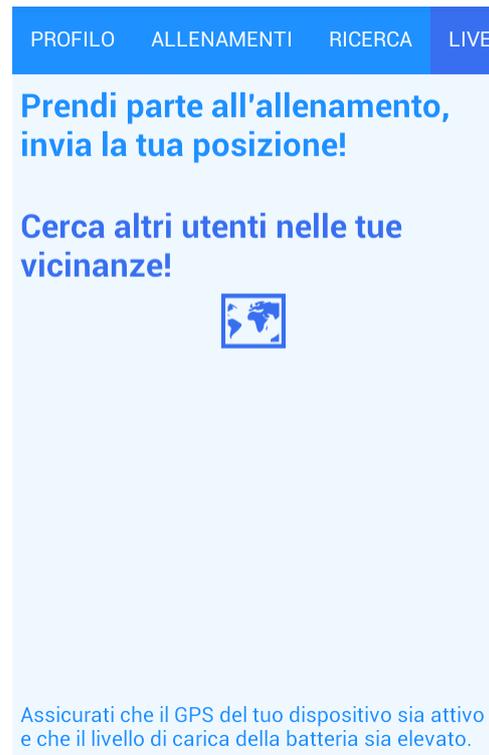
È bene precisare che le activity di ricerca allenamenti e di ricerca utenti sono completamente separate fra loro ed avviabili distintamente dai due pulsanti di avvio ricerca presenti nelle relative sezioni.

3.11 Sezione Live

Durante l'implementazione di questa fase sono saltati all'occhio alcuni aspetti che non si erano notati in fase di progettazione. Nella fase precedente si era infatti pensato di suddividere questa sezione in due parti. La prima parte relativa all'invio della posizione geografica dell'utilizzatore, la seconda relativa alla visualizzazione degli utenti su di una mappa. La scelta di implementare il servizio in questo modo è però sembrata abbastanza frammentaria ed inutile, specialmente alla luce del fatto che il segnale GPS sarebbe stato da attivare anche all'interno dell'activity contenente la mappa.

Si è deciso perciò di implementare un'unica soluzione: sia il servizio di geolocalizzazione che il servizio di visualizzazione della mappa vengono forniti dall'activity della mappa. Questo aspetto comporta quindi diverse modifiche alle scelte effettuate fino ad ora per la sezione live.

Viene innanzitutto modificato il layout presente nella sezione principale nel modo indicato alla pagina seguente.



Viene inoltre leggermente modificato il senso della sezione: se, infatti, nella versione progettata in precedenza il servizio consentiva di visualizzare gli utenti attivi, questa versione è più orientata a visualizzare gli utenti che stanno consultando la mappa, cioè quegli utenti che stanno cercando compagni di allenamento a cui unirsi. Si aumenta quindi la possibilità di trovare qualcuno veramente interessato a pedalare in compagnia nelle vicinanze.

L'apportare di questa modifica ha mostrato un unico possibile difetto: occorre tenere l'activity di visualizzazione perennemente attiva nel caso si volesse pedalare fornendo la propria posizione. Alla luce di quanto detto poco fa in ogni caso è sembrata un rischio abbordabile in quanto gli utenti visualizzati sulla mappa saranno prevalentemente coloro che ricercheranno dei compagni a cui unirsi e, una volta individuati, provvederanno a chiudere la mappa e a raggiungerli, riaprendola nel caso volessero ricontrollarne le posizioni.

Questa modifica ha ridefinito anche due servizi sul server: all'interno del servizio *ottieniVisualizzabili.php* è stato inglobato anche il servizio *inviaCoordinate.php*, che è quindi stato rimosso dal server.

3.11.1 Ottenimento della posizione

Per evitare problemi futuri si è deciso di inserire un controllo all'avvio dell'activity: se il segnale GPS non risulta attivo l'activity si arresta immediatamente e viene mostrato un messaggio di errore sotto forma di Toast, ovvero un messaggio che scompare automaticamente e permette all'utente di svolgere qualsiasi azione sul dispositivo nel frattempo.

FASE DI IMPLEMENTAZIONE

Se l'activity viene avviata in maniera corretta viene attivato un ascoltatore GPS, ovvero un componente che permette di rilevare le coordinate geografiche dell'utente. Si è deciso di compiere la rilevazione ogni secondo per renderla il più preciso possibile.

Una volta che l'utente viene rilevato l'ascoltatore provvede a comunicare, ovviamente sempre attraverso un apposito task parallelo, la posizione dell'utente al server e ad ottenere tutti gli utenti che stanno utilizzando il servizio live, come già descritto nella spiegazione del servizio *ottieniVisualizzabili.php*.

Quando l'utente dovesse decidere di terminare l'activity l'ascoltatore viene rimosso e si accede al servizio *pulisciCoordinate.php* per reimpostare la Latitudine e la Logitudine dell'utente a NULL.

Osservando, tramite PhpMyAdmin, i valori in database ci si è accorti che le coordinate (inserite come double) venivano troncate, perdendo informazioni significative relativamente alla posizione. I due campi della tabella Utente relativi a Latitudine e Longitudine sono stati quindi modificati da double a stringa, ovvero sono stati resi di tipo char(50).

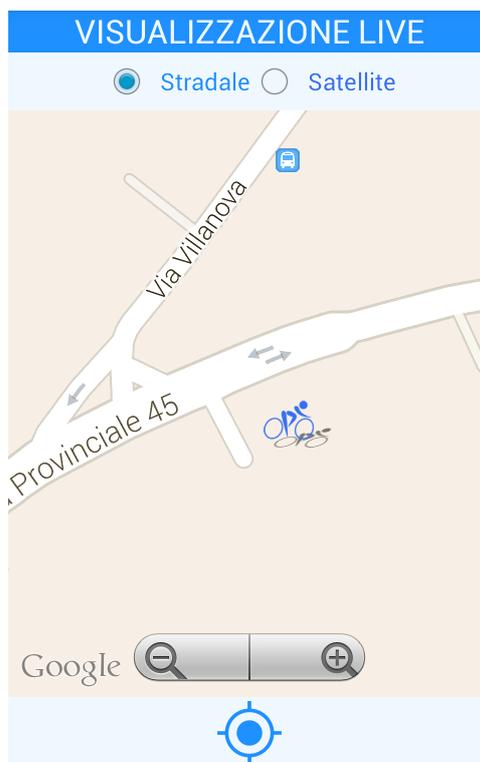
3.11.2 Modifiche alla visualizzazione della mappa

Per realizzare la rappresentazione della mappa è stata utilizzata la risorsa messa a disposizione dalle librerie online di Google, denominata MapView. Il componente verrà quindi scaricato dalla rete ogni volta che dovrà essere rappresentato ed il caricamento delle sue informazioni interne (la cartina) avviene sempre attraverso un caricamento di dati online.

Per rendere semplice e intuitiva la mappa si è deciso di apportare alcuni accorgimenti grafici. Si è deciso di fornire la possibilità di far scegliere all'utente la visualizzazione della mappa che preferisce, potendo scegliere fra cartina stradale oppure rappresentazione satellitare.

All'avvio dell'activity la mappa viene centrata e viene effettuato uno zoom automatico sulla posizione dell'utilizzatore, zoom che potrà essere regolato tramite appositi controlli forniti dalla mappa stessa. È stato inoltre inserito un pulsante in grado di ripetere l'operazione di centramento e zoom sull'utilizzatore in qualunque momento.

Il nuovo layout visualizzato nell'activity della mappa è rappresentato nella pagina seguente.



3.11.3 Rappresentazione degli utenti nella mappa

Va introdotta in ultimo, relativamente alla visualizzazione della mappa, la descrizione di come gli utenti vengono effettivamente rappresentati in essa.

Si è scelto di rappresentare gli utenti utilizzando la stessa immagine di ciclista stilizzato che è stata adoperata per l'icona dell'applicazione. L'icona verrà visualizzata con due possibili colori, già introdotti in fase di progettazione: l'utilizzatore verrà rappresentato in blu, gli utenti generici verranno rappresentati in celeste. Questa scelta è stata fatta per rendere sempre chiara la posizione dell'utilizzatore rispetto agli altri utenti. Le icone degli utenti verranno inoltre ripulite e ricaricate ad ogni nuova iterazione dell'ascoltatore e la mappa verrà invalidata automaticamente per permettere l'aggiornamento automatico dei suoi dati.

Una volta visualizzati gli utenti si è deciso inoltre di fornire la possibilità di visualizzarne il nome ed il cognome. Una volta premuta l'icona di un ciclista si aprirà quindi un dialog contenente questi ultimi due dati. Si è deciso di rappresentare i dati in questo modo e non direttamente sulla mappa per non rendere troppo pesante il layout, saturandolo con dati ottenibili anche attraverso altri strumenti, i dialog appunto.

Per evitare un'ulteriore appesantimento al caricamento non vengono effettuate restrizioni ai soli utenti nelle vicinanze, ma vengono visualizzati tutti gli utenti attivi. Nel caso in cui il numero di utenti iscritti all'applicazione dovesse essere più elevato del previsto si effettueranno ulteriori riconsiderazioni in merito.

3.12 Gestione delle Notifiche

L'ultimo componente tenuto in considerazione in fase di implementazione è il gestore delle notifiche. Questo componente, come già visto in fase di progettazione layout, fornisce alcune possibilità di gestione all'utente nella sezione impostazioni.

Il gestore delle notifiche è stato implementato in una classe denominata *GestoreNotifiche.java*, se ne analizzerà ora il comportamento.

3.12.1 Introduzione al servizio

La classe che implementa il gestore delle notifiche estende la classe *Service*. Che cos'è un service? Un service è un componente software che consente di eseguire delle operazioni in background senza dover interagire con l'utente, che nel frattempo può dedicarsi ad altre attività, interne o esterne all'applicazione. Va specificato inoltre che un *Service* fa comunque parte del thread principale, a differenza di quanto si era visto per gli *AsyncTask*.

Per questo motivo il servizio di gestione delle notifiche non fa altro che lanciare al suo interno due thread separati, che gestiscono diversi tipi di notifiche. All'interno di questi thread sarà quindi possibile eseguire tutte le operazioni di connessione al server per ottenere i dati delle notifiche.

3.12.2 Avvio del servizio

Il gestore delle notifiche viene avviato in automatico all'avvio del dispositivo. Questa caratteristica è stata implementata utilizzando una classe di appoggio chiamata *BootReceiver.java*. Quest'ultima classe è molto semplice e viene fornita qui di seguito:

```
public class BootReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED)){
            Intent serviceIntent=new Intent(context,GestoreNotifiche.class);
            context.startService(serviceIntent);
        }
    }
}
```

BootReceiver.java implementa un ricevitore, ovvero un componente in grado di mettersi in ascolto di eventi particolari che avvengono nel sistema Android. In questo caso il ricevitore è in grado di riconoscere quando il dispositivo termina la fase di boot e quindi avvia successivamente il servizio di gestione delle notifiche. Per rendere funzionante questo componente è necessario inserire all'interno del file *manifest* la seguente dicitura:

```
<service android:name="GestoreNotifiche">
    </service>
    <receiver android:name="BootReceiver" android:enabled="true"
        android:exported="false">
        <intent-filter>
```

FASE DI IMPLEMENTAZIONE

```
<action android:name="android.intent.action.BOOT_COMPLETED"/>
</intent-filter>
</receiver>
```

Si è deciso di specificare questi aspetti del codice in quanto non sono stati immediatamente banali da implementare, ma hanno comportato una ricerca approfondita in rete siccome si trattava di una tematica mai approfondita.

Nel caso in cui il servizio dovesse interrompersi per qualche motivo imprevisto si è deciso di controllare il suo stato ogni volta che l'utente avvia l'applicazione: nel caso in cui il servizio non dovesse essere attivo si provvede immediatamente a ripristinarlo. Anche in questo caso la funzione implementata nella main activity non si è rivelata subito banale, per cui si è scelto di riportarle qui di seguito:

```
/**
 * Funzione che verifica se il servizio è attivo e se non lo è lo attiva
 */
private void MyServiceRunning() {

    boolean attivo=false;

    ActivityManager manager = (ActivityManager) getSystemService(ACTIVITY_SERVICE);
    for (RunningServiceInfo service : manager.getRunningServices(Integer.MAX_VALUE)){
        if("christian.gamberini.ciclismomobile.GestoreNotifiche".equals(service.getService().getClassName())) {
            //Il servizio è attivo
            attivo=true;
            break;
        }
    }

    if(!attivo){
        //Se non è attivo lo devo attivare
        Intent serviceIntent=new
Intent(getApplicationContext(),GestoreNotifiche.class);
        getApplicationContext().startService(serviceIntent);
    }
}
```

3.12.3 Funzionamento del servizio

Una volta avviato il servizio vengono lanciati due thread. Il primo si occupa di gestire le notifiche relative agli allenamenti creati dall'utente, il secondo si occupa di fornire il promemoria di inizio di un allenamento a cui si risulta iscritti.

Ciascuno dei due thread implementa al suo interno un ciclo infinito in cui vengono eseguite alcune operazioni comuni: ogni thread resta in riposo per un certo periodo di tempo, una volta attivo controlla se il dispositivo è connesso alla rete e, nel caso lo fosse, provvede a controllare se la notifica in esame è stata abilitata o meno dal pannello delle impostazioni. Per eseguire questo ultimo passaggio sono stati salvati nelle preferenze tre valori booleani, rappresentanti l'eventuale abilitazione di un tipo di notifica. Al caricamento della pagina principale i CheckBox corrispondenti nel layout delle impostazioni verranno settati al valore presente nelle preferenze.

Si passa ora ad analizzare il comportamento specifico dei due thread:

3.12.3.1 Thread di gestione degli allenamenti organizzati

In questo thread viene implementato il riconoscimento di due tipi di notifiche: quando un altro utente commenta un allenamento creato dall'utilizzatore e quando un utente partecipa ad un allenamento organizzato dall'utilizzatore.

Se il primo tipo di notifica descritta è abilitato il servizio richiede al server prima l'elenco degli allenamenti organizzati dall'utilizzatore (si è reso necessario scrivere un nuovo servizio nel server *ottieniOrganizzati.php* per ottenere l'elenco ed i dati degli allenamenti organizzati da un utente), poi per ognuno di essi l'id dell'ultimo commento inserito dagli utenti partecipanti. Nelle preferenze è poi stata inserita una variabile intera, rappresentante l'id dell'ultimo commento effettuato dagli utenti in uno degli allenamenti dell'utilizzatore. Ogni volta che il server ritorna un'id di ultimo commento per un allenamento questo valore viene confrontato con il numero presente nelle preferenze: nel caso il valore ottenuto fosse maggiore viene inviata una notifica. Una volta iterati tutti gli allenamenti organizzati il valore dell'id commento più alto ottenuto viene registrato nelle preferenze. Per rendere funzionante questo processo si è dovuto implementare un nuovo servizio sul server denominato *getIdUltimoCommento.php*, che prende in ingresso la mail dell'utente utilizzatore e l'id dell'allenamento di cui si vuole ottenere l'ultimo commento registrato dagli utenti. I dati vengono ovviamente reperiti dalla tabella *Commento*.

Per quanto riguarda il secondo tipo di notifiche si è scelto di operare allo stesso modo. È stata quindi modificata la struttura della tabella *Partecipa*, inserendo il campo *Conteggio* di tipo intero auto incrementante. In questo modo realizzando il servizio sul server *getIdUltimaPartecipazione.php* ed inserendo una nuova preferenza è possibile gestire questo tipo di notifiche in maniera identica alla precedente. È bene specificare che l'elenco degli allenamenti organizzati viene richiesto comunque una sola volta e viene utilizzato per la gestione di entrambe le notifiche.

Una volta visualizzate le notifiche (contenenti un messaggio generico a seconda del loro tipo) l'utente potrà selezionarle per aprire una scheda: nel caso della prima notifica verrà aperta la scheda dei commenti dell'allenamento in questione, nel caso della seconda notifica verrà aperta semplicemente la scheda allenamento, da cui poi si potrà accedere alla lista di tutti gli iscritti.

Il thread una volta completate le operazioni rimane inattivo per due secondi.

3.12.3.2 Thread di promemoria

In questo thread viene implementato il riconoscimento del terzo ed ultimo tipo di notifiche: il promemoria di inizio di un allenamento a cui si risulta iscritti.

Per implementare questo riconoscimento è stata inserita una nova pagina sul server, *ottieniImminenti.php*: da questa pagina viene ritornato un elenco di allenamenti a cui l'utente passato come parametro risulta iscritto che avranno inizio entro due ore.

FASE DI IMPLEMENTAZIONE

Per evitare il ripetersi continuo di una notifica già ricevuta anche in questo caso si è deciso di operare come per le precedenti due tipologie di notifiche: viene gestito dalle preferenze l'id dell'ultimo allenamento notificato e viene confrontato con gli id degli allenamenti imminenti ricevuti dal server.

Dalla notifica visualizzata sarà possibile accedere direttamente alla scheda dell'allenamento di interesse.

Siccome non si è scelto di vietare l'iscrizione a più allenamenti in contemporanea si potranno ricevere notifiche di allenamenti imminenti in contemporanea, sta negli utenti controllare a quali allenamenti si risulta iscritti.

Questo thread una volta completate le operazioni rimane inattivo per mezz'ora, non si è ritenuto infatti sensato eseguire controlli in un intervallo di tempo inferiore per prevenire traffico dati ed utilizzo della batteria che sarebbero risultati inutili.

FASE DI IMPLEMENTAZIONE

4. OSSERVAZIONI CONCLUSIVE

Terminata la fase di implementazione è giunto il momento di eseguire alcuni test sul prodotto ottenuto e compiere alcune osservazioni su lavoro finale.

4.1 Osservazioni sui Test Effettuati

Analizziamo in questo punto le osservazioni derivanti dai test effettuati direttamente su alcuni dispositivi.

4.1.1 Errori di esecuzione

Dopo aver eseguito alcuni test il problema principale riscontrato è stato relativo alla gestione del servizio delle notifiche. Per la precisione ci si è accorti del problema testando l'invio delle notifiche su due dispositivi differenti presenti nella stessa stanza. Alle volte quando si inseriva un nuovo commento infatti l'utente organizzatore dell'evento non riceveva nessuna notifica, pur avendola abilitata dalle impostazioni. È stato problematico riscontrare il problema in quanto l'invio delle notifiche è sempre avvenuto in modo corretto sin dalle prime fasi di implementazione. Andando però ad osservare lo stato dei servizi attivi nei dispositivi ci si è accorti che il servizio di notifiche non era funzionante, per questo la notifica non veniva ricevuta; riavviando quindi il servizio dall'applicazione infatti la notifica veniva immediatamente consegnata correttamente. Si è quindi dovuto ricercare il problema di questa interruzione imprevista. Dopo vari tentativi ed osservazioni si è notato che il servizio veniva interrotto durante il passaggio di connettività fra Wi-Fi e connessione dati o viceversa. Osservando quindi le informazioni di log e forzando l'errore si è notata la presenza di un'eccezione del tipo `NullPointerException`. Ispezionando quindi il codice si è finalmente individuato il problema: nella funzione della classe `ServerCiclismoMobile.java` adibita ad ottenere il risultato della connessione ad un url sul server veniva ritornato il valore NULL qualora vi fossero dei problemi di connessione durante questa fase. Il valore NULL veniva poi controllato come una stringa dalle funzioni specifiche dei servizi, generando appunto l'eccezione e la relativa interruzione del servizio. Per risolvere il problema si è quindi fatto in modo che la funzione, nel caso dovesse presentarsi un qualsiasi tipo di problema durante la sua esecuzione, ritorni il valore "0" al posto di NULL, come se l'errore fosse stato generato dal server. Implementando questa soluzione il problema è stato completamente risolto.

Non sono stati rilevati ulteriori problemi significativi causanti l'arresto dell'applicazione o del suo servizio associato.

4.1.2 Reattività dei servizi

Le connessioni al server si dimostrano sempre abbastanza veloci anche in presenza di un segnale internet abbastanza debole. Anche quando si devono scaricare dati di

OSSERVAZIONI CONCLUSIVE

allenamenti che risultano abbastanza complessi, il tempo di esecuzione medio è abbastanza basso.

La reattività del servizio di notifiche è risultata ottima: si devono attendere al massimo un paio di secondi, dovuti all'interruzione momentanea del thread. Nella maggior parte dei casi la notifica viene consegnata quasi immediatamente alla registrazione del dato sul server.

L'unico servizio di cui non si è rimasti propriamente soddisfatti in termini di reattività è legato alla visualizzazione della mappa. Si è infatti rilevato un leggero ritardo nell'aggiornamento della mappa. Siccome questo ritardo non è comunque degno di una particolare attenzione si è deciso di non apportare modifiche al servizio.

Le ultime osservazioni fornite sono relative al traffico dati e all'utilizzo della batteria.

Il traffico dati maggiore viene registrato quando le notifiche sono state abilitate, questo è dovuto al fatto che viene ogni volta scaricato un elenco degli allenamenti. Per quanto riguarda invece l'utilizzo della batteria non si sono rilevati cambiamenti considerevoli nella durata della carica. L'applicazione a fine giornata consuma mediamente il 4% della batteria, questo dato dipende ovviamente dalla frequenza di utilizzo del servizio.

La visualizzazione live ovviamente merita un discorso a parte in quanto le richieste del GPS e le connessioni ad internet (dovute anche alla visualizzazione della mappa) causano un notevole aumento sia del traffico dati sia dell'utilizzo della batteria, come avviene in ogni caso in servizi simili come ad esempio nelle applicazioni di navigazione stradale.

4.2 Proposti Futuri

In conclusione al lavoro si reputa necessario fornire alcune considerazioni su ciò che sarà possibile effettuare in futuro nel progetto.

4.2.1 Risoluzione bug minori

Potrebbero essere presenti alcuni problemi non rilevati durante la prima fase di test. Nel caso in cui si volesse commercializzare l'applicazione sarà necessario richiedere un feedback costante agli utenti per poter aggiornare al meglio il lavoro svolto. È necessario precisare che le fasi di test condotte fino ad ora hanno chiamato in causa un numero molto ridotto di utenti utilizzatori.

4.2.2 Riduzione traffico dati

Nel caso in cui si volesse mettere in commercio l'applicazione per un numero elevato di utenti sarà necessario ottimizzare al meglio le connessioni al server, per far transitare dalla rete il minor numero di dati possibili. Andrà quindi ottimizzato il

OSSERVAZIONI CONCLUSIVE

servizio di notifiche, magari tramite l'inserimento di una tabella apposita in database in cui memorizzare le notifiche non ancora lette dall'utente cui sono indirizzate.

4.2.3 Scelta di un dominio appropriato

Nel caso in cui si notasse un'iscrizione elevata di utenti il dominio gratuito potrebbe non essere più sufficiente a causa di eventuali rallentamenti e politiche di protezione attuate dal servizio Altvista. Occorrerà quindi ponderare accuratamente l'acquisto di un dominio web che consenta di eseguire tranquillamente tutte le operazioni necessarie al buon funzionamento dell'applicazione.

4.2.4 Traduzione in più lingue

Attualmente l'applicazione è stata programmata solamente in lingua italiana. Nel caso in cui vi fosse necessita di estenderne un utilizzo all'estero sarà necessario fornire una traduzione in altre lingue, quantomeno in lingua inglese. Per risolvere questo problema Android offre un potentissimo sistema di gestione delle stringhe. È stato infatti generato un file in cui sono presenti tutte le stringhe utilizzate dall'applicazione. Sarà sufficiente quindi duplicare e tradurre il seguente file per ogni lingua che si vorrà inserire, verrà quindi tradotta agli occhi dell'utilizzatore tutta l'applicazione in base alla lingua impostata come predefinita nel dispositivo. Un utente inglese ad esempio visualizzerà l'applicazione in inglese, senza dover modificare alcuna impostazione all'interno dell'applicazione. Nel caso in cui la lingua del dispositivo non venisse rilevata fra quelle disponibili per l'applicazione verrà scelta la lingua italiana come lingua di default.

Ovviamente i dati presenti in database sono impossibili da tradurre.

4.2.5 Ottimizzazione grafica

L'ultimissimo accorgimento da adoperare nel caso in cui si volesse inserire in commercio l'applicazione è quello di ottimizzare le viste grafiche per tutti i dispositivi disponibili. Attualmente i layout dell'applicazione non hanno causato problemi sui pochi dispositivi su cui sono stati testati, occorrerà comunque eseguire una serie di test molto più approfondita.

A livello grafico inoltre si potrebbero apportare diverse modifiche per rendere l'applicazione più accattivante ed intuitiva all'uso. Per questa operazione sarà però necessario contattare degli esperti in grafica.

4.2.6 Implementazione in altri sistemi operativi

Per poter aumentare il numero di utenti iscritti sarà inoltre fondamentale in futuro attrezzarsi per poter implementare l'applicazione nei due sistemi operati scartati in fase di implementazione: iOS e Windows Phone. Siccome l'implementazione in questi sistemi richiederebbe un investimento economico è da valutare anche la possibilità di poter trarre alcuni guadagni dall'applicazione.

OSSERVAZIONI CONCLUSIVE

I casi possibili presi in considerazione per ottenere un guadagno sono due:

- Realizzare una versione a pagamento completa ed una versione gratuita con alcune limitazioni.
- Realizzare due versioni, entrambe complete: la versione gratuita contenente banner pubblicitari e la versione a pagamento libera.
- Ottenere degli sponsor, magari nell'ambito dello sport scelto, per rendere l'applicazione completa e gratuita per tutti.

Al momento attuale se si dovesse compiere una scelta si sarebbe comunque intenzionati a seguire la seconda strada, più semplice ed abbordabile, anche grazie agli strumenti di Google AdSense, che consentono l'inserimento di banner pubblicitari automatici nell'applicazione.

BIBLIOGRAFIA

(s.d.). Tratto il giorno Novembre 14, 2013 da [stackoverflow.com](http://stackoverflow.com/questions/18673151/android-activity-contentview-size):
<http://stackoverflow.com/questions/18673151/android-activity-contentview-size>

Ferretti, M. (1949, Giugno 10). <<Un uomo solo è al comando; la sua maglia è biancoceleste; il suo nome è Fausto Coppi.>>. (t. t. Frase pronunciata all'apertura della radiocronaca della Cuneo-Pinerolo, Intervistatore)

RINGRAZIAMENTI

A mia madre, la donna più straordinaria della Terra, sempre pronta a fare sacrifici per aiutarmi ad andare avanti.

Ai miei nonni Ida e Aldino che mi hanno cresciuto e senza i quali probabilmente questo lavoro non sarebbe mai potuto essere compiuto.

A Benedetta, che mi ha semplicemente cambiato la vita e mi ha sopportato nei momenti più duri di studio e di lavoro.

A Giardo e Beto, compagni di studio e di delirio, che sono diventanti più due fratelli per me.

A tutti i miei amici che sento sempre vicini e a tutti coloro che mi hanno aiutato a scrivere questo documento e a testare l'applicazione.

L'ultimo ringraziamento va poi alla mia bicicletta, fonte di ispirazione e mezzo di sfogo in tutti i momenti di stress fisico e psicologico.