Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Informatica

SVILUPPO APPLICAZIONE ANDROID TALEMOTION

Laureando

Antonio Davide Calì

Relatore

Prof. Giorgio Maria Di Nunzio

Padova, 26 Settembre 2013

Alla mia famiglia e a me stesso...

Sommario

Lo scopo del presente documento è quello di illustrare le attività svolte durante lo stage accademico svolto dal 10/07/2013 al 02/09/2013 per una durata complessiva di 250 ore lavorative.
L'obiettivo dello stage è quello di ideare e sviluppare l'applicazione Android del sito web www.talemotion.com.
Questo documento descrive tutte le motivazioni e le scelte intraprese per portare a termine l'obiettivo fissato; in particolare si propone un percorso che porta, dallo studio degli strumenti disponibili e dei meccanismi di funzionamento di essi stessi, alla risoluzione del problema seguendo ordinatamente le seguenti attività:

- Analisi dei requisiti, per definire le richieste del cliente;
- Studio di fattibilità, per comprendere i funzionamenti e i metodi necessari per il compimento del lavoro;
- Sviluppo software, per dar una spiegazione generale di come l'applicazione sia stata progettata;

Indice

Capitolo 1

Introduzione

Il progetto svolto durante l'attività di stage riguarda lo sviluppo dell'applicazione Android per il sito web www.talemotion.com. Sono state svolte diverse attività, dall'analisi di requisiti, allo studio di framework, fino alla progettazione software, tutte finalizzate allo sviluppo dell'applicazione stessa.

Ogni attività viene trattata nelle sezioni successive così suddisive:

- Capitolo 1: Introduzione
- Capitolo 2: Analisi dei requisiti
- Capitolo 3: Sviluppo
- Capitolo 4: Conclusioni

In questo capitolo si vogliono dare alcune informazioni generali riguardo lo scopo del tirocinio effettuato, l'azienda EITEAM, il sito web talemotion.com e l'analisi che ha portato lo sviluppo dell'applicazione sul sistema operativo Android.

1.1 Scopo dello stage

Lo scopo dello stage è stato lo sviluppo di un'applicazione su ambiente Android del sito www.talemotion.com. Lo sviluppo ha avuto come primo obiettivo quello di rendere fruibile un'applicazione user friendly che permettesse la lettura e la recensione dei libri presenti sul sito.
Il lavoro oggetto dello stage si è basato sui seguenti punti:

- analisi degli strumenti e delle tecnologie utilizzate dall'azienda
- analisi del sito web www.talemotion.com;
- analisi dei framework utilizzati per la realizzazione del sito
- analisi database adoperati dal sito

- decisioni grafiche riguardanti l'applicazione
- organizzazione gerarchica delle features da implementare
- finalizzazione dell'applicazione

Le prime fasi, di studio, individuano la fattibilità del progetto in accordo con le tecnologie in uso per www.talemotion.com, gli standard in esso utilizzati, con le tecnologie disponibili. Ciò porterà ad avere piena conoscenza delle possibilità di utilizzo di strutture già esistenti sul sito o la decisione di scrivere ex novo algoritmi necessari per l'implementazione futura dell'applicazione, tutto finalizzato alla possibilità di accedere, aggiornare e aggiungere dati presenti sul database.

1.2 Descrizione dell'azienda

EITEAM è una società cooperativa sociale, ovvero un particolare tipo di società cooperativa, di tipo B, cioè che svolge attività di diverso genere (agricole, industriali, commerciali, di servizi) al fine di inserire persone svantaggiate (ad esempio: invalidi, alcolisti, tossicodipendenti, minori in età lavorativa con difficili situazioni familiari) nel mondo del lavoro. L'azienda propone i seguenti servizi:

- inserimento dati;
- archiviazione documentale;
- siti Internet;
- sviluppo software;
- gestione mail e hosting;
- progettazione grafica;
- assistenza pc e reti.

Oltre a software su commissione, fornisce i seguenti servizi:

- Cariddi, per la gestione degli incassi con scadenza periodica;
- VPOS, erogato dall'azienda assieme a Banca Popolare Etica, per raccogliere fondi o pagamenti di servizi mediante carta di credito;
- WEBLOOK, un cms proprietario per la creazione e gestione di siti web personalizzati.

La scelta di questa azienda per la realizzazione del progetto di stage è motivata dalla possibilità di entrare a far parte di un gruppo di professionisti giovani e capaci, dotati di svariate competenze in ambito Web all'interno di un ambiente ricco e stimolante che offre uno stile di lavoro e formativo incentrato su eticità e rispetto della pari dignità di ogni persona.

1.3 TaleMotion

Talemotion è il nome dato al sito web oggetto di integrazione e analisi di questa tesi.

Il sito nasce dall'idea di portare uno scrittore, autore di libri, nella rete Internet: tramite questo sistema un utente può generare un elaborato e stabilire se mantenerlo privato (cioè visibile solamente a se stesso) oppure pubblicarlo (gratuitamente), rendendolo visibile agli altri utenti (che fungono da lettori). In questo senso Internet contribuisce in modo significativo alla produttività di uno scrittore, permettendogli di raggiungere in poco tempo un vasto numero di utenti, offrendo maggiore visibilità dei propri elaborati rispetto alla normale pubblicazione.
Nel progetto è stato inoltre valutato anche l'aspetto economico: l'obiettivo è quello di stabilire un collegamento diretto tra l'autore e i suoi lettori, tramite un meccanismo di vendita diretta online (e-commerce) attraverso il quale un utente, interessato a un racconto, o per passaparola o in seguito alla lettura dell'abstract (un testo introduttivo che pubblicizza ciò che il lettore si appresta a leggere proseguendo nella visualizzazione), decide di acquistarlo: questa azione, nel contesto di Talemotion, comporta la possibilità di visualizzarlo, di stamparlo o di scaricarlo, acquisendone quindi i diritti di lettura. Esiste quindi una distinzione per le pubblicazioni: un'opera pubblicata può essere gratuita, quindi leggibile senza distinzioni da tutti gli utenti del sito, oppure a pagamento, cioè visibile solamente agli utenti che effettuano l'acquisto virtuale del racconto.

Il sito web www.talemotion.com è stato sviluppato da EITEAM (Sezione 1.2) per commissione di *ARS Network S.r.l.*

1.4 Android

Android, l'ormai noto sistema operativo firmato Google, presenta molteplici aspetti e sfaccettature.
In questa sezione verrà proposta una breve introduzione ad Android e alle motivazioni che hanno fatto ricadere lo sviluppo dell'applicazione lungo i binari di questa piattaforma.

1.4.1 La nascita di Android

Android nasce nel 2003 in California, da Andy Rubin, Rich Miner e Chris White, i quali si uniscono con lo scopo di creare dei dispositivi più "intelligenti" di quanto non fossero quelli dei cellulari dell'epoca.
Lavorando segretamente ad un progetto riguardo software per cellulari, nel 2005 Google ingloba Android Inc., lasciando tuttavia a capo del progetto Rubin, Miner e White.
Due anni più tardi, nel 2007, Google richiede la registrazione di vari brevetti riguardanti tecnologie software molto vicine alla tecnologia mobile, stringe inoltre accordi commerciali con i più grandi produttori e con le più grandi aziende operanti nel settore della telefonia mobile (HTC, Samsung e T-Mobile) e della produzione hardware (Qualcomm e Texas Instruments), alimentando le già presenti voci riguardo un eventuale ingresso nel settore mobile della compagnia di Mountain View.
Le voci trovano veridicità nel 2008, quando Google svela il suo progetto presentando il

primo smartphone Android, l'HTC Dream.
Il fatto che Android sia distribuito in formato open-source, sotto licenza Apache, porta - nei cinque anni a seguire - ad una enorme diffusione della piattaforma. Sebbene Android sia infatti nato per dispositivi mobili touchscreen, potendo essere liberamente distribuito e modificato ha esteso la sua diffusione ai più svariati dispositivi, quali ad esempio televisioni e console videoludiche.
È tuttavia nel mondo degli smartphone che Android detiene il record per il sistema operativo più diffuso, con più del 75% di market share, grazie anche al supporto fornito da Google alle migliaia di developers che vogliono cimentarsi nella creazione di app; applicazioni che estendono le funzionalità di un dispositivo Android, come quella che verrà presentata alla fine di questa tesi.

1.4.2 Perché Android?

Perché si e quindi deciso di utilizzare Android? Le ragioni che hanno portato a questa scelta sono molteplici: in primis l'enorme diffusione a livello mondiale di Android: Android ha avuto una grande ascesa nell'ultimo anno. La recente indagine di IDC[1] ha rivelato che il market share controllato da Android è ora al 79,3 per cento, una crescita di più del 10 per cento rispetto al 69,1 del secondo trimestre del 2012. Mentre al secondo posto della classifica degli OS si piazza Apple iOS con il 13,2 per cento, in declino rispetto al 16,6 dell'anno scorso.
Seconda ragione da considerare la sua licenza Apache che comporta l'essere completamente open-source, oltre a una facile reperibilità delle API android e ottimi tutorial e guide reperibili online che introducono agevolmente e rapidamente alla programmazione nel mondo Android.
Un altro punto a favore di Android è la possibilità di poter scrivere le applicazioni utilizzando Eclipse, un IDE anch'esso open-source ufficialmente supportato da Google tramite plugin appositamente sviluppati, che rende semplice lo studio e lo sviluppo di una applicazione su questa piattaforma.
Uno degli aspetti più importanti inoltre risiede nel fatto che, nonostante l'intero sistema operativo sia basato su Linux, le applicazioni livello utente siano scritte primariamente in Java.
La programmazione per Android inoltre risulta essere gratuita, non necessita l'acquisto di un kit di sviluppo né di una licenza per la stessa; la distribuzione sul Play Store[2] è inoltre più libera, più veloce e meno restrittiva rispetto alla concorrenza quale iOS.
Android ha reso possibile, dato il costo relativamente contenuto rispetto ai concorrenti (Apple in primis), l'accesso agli smarthphone molto più economico per gli utenti e allo stesso tempo ha permesso/obbligato chi lavora nel campo della programmazione di applicazioni a doversi confrontare con esso (e a guadagnarci). Infine l'indubbia solidità dell'azienda Google che rende Android un sistema stabile, in continuo sviluppo e che difficilmente cadrà in disuso e declino in un breve periodo.

[1] International Data Corporation: http://www.idc.com/
[2] Market Place della piattaforma Android: https://play.google.com/store

Top Smartphone Operating Systems, Shipments, and Market Share, 2013 Q3 (Units in Millions)

Operating System	2Q13 Unit Shipments	2Q13 Market Share	2Q12 Unit Shipments	2Q12 Market Share	Year-over-Year Change
Android	187.4	79.3%	108	69.1%	73.5%
iOS	31.2	13.2%	26	16.6%	20.0%
Windows Phone	8.7	3.7%	4.9	3.1%	77.6%
BlackBerry OS	6.8	2.9%	7.7	4.9%	-11.7%
Linux	1.8	0.8%	2.8	1.8%	-35.7%
Symbian	0.5	0.2%	6.5	4.2%	-92.3%
Others	N/A	0.0%	0.3	0.2%	-100.0%
Total	**236.4**	**100.0%**	**156.2**	**100.0%**	**51.3%**

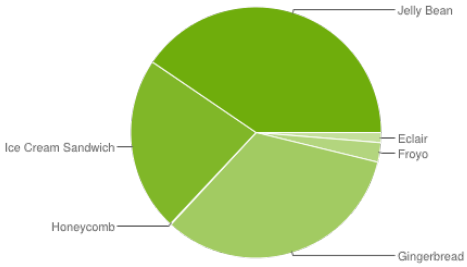
Figura 1.1: Tabella comparativa vendita device smartphone prodotta da IDC

1.4.2.1 Versione Android

Per lo sviluppo dell'applicazione è stata scelta come versione minima la 4.0 Ice Cream Sandwitch con API versione 14.
La scelta è ricaduta su questa versione per motivazioni prettamente grafiche.
Data ormai la grande diffusione di quest'ultima versione, non si ritiene essere restrittiva la scelta, in quanto, più di un device su due supporta le API necessarie.

Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.2%
2.2	Froyo	8	2.5%
2.3 - 2.3.2	Gingerbread	9	0.1%
2.3.3 - 2.3.7		10	33.0%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	22.5%
4.1.x	Jelly Bean	16	34.0%
4.2.x		17	6.5%

(a) Tabella riassuntiva delle distribuzioni Android



(b) Grafico a torta illustrativo delle distribuzioni Android

Figura 1.2: Dati sulla distribuzione delle versioni Android rese note da Google stessa aggiornata all'01 Agosto 2013: http://developer.android.com/about/dashboards/index.html

Capitolo 2

Analisi dei requisiti

Il capitolo proposto vuole trattare e descrivere i requisiti imposti e/o concordati dallo studente, dall'azienda e dall'ambiente in cui si sarebbe definita e sviluppata l'applicazione.
Si vuole dare un'analisi generale delle funzionalità e moduli necessari allo sviluppo, sviluppo che verrà trattato nel capitolo successivo (Capitolo 3).

2.1 Specifica dei requisiti

L'analisi di Talemotion, effettuata assieme ai principali membri dell'azienda ha fatto emergere i requisiti obbligatori che le attività di stage dovranno soddisfare.
La classificazione dei requisiti e la loro descrizione è di seguito riportata.

2.1.1 Requisiti funzionali (F)

F1. Modulo lettura

F2. Download eBook

F3. Scheda di un'opera

F4. Ricerca di un'opera per titolo

F5. Ricerca avanzata di un'opera

F6. Sezione Leggi

F7. Consultare le recensioni di un'opera

F8. Recensire e valutare un'opera

F9. Login utente con memorizzazione delle credenziali

F10. Registrazione nuovo utente

F11. Homepage applicazione dinamica

F12. Condivisione via Social Network

2.1.2 Requisiti di qualità (Q)

Q1. Homepage dell'applicazione simile a Homepage sito web.

Q2. Modulo lettura provvisto di copertina

Q3. Menù laterale per accesso a impostazioni

Q4. Bottom Bar per la navigazione tra sezioni

Q5. Scheda libro seguenti informazioni: immagine di copertina, titolo, sottotitolo, autore, genere, sottogenere, prezzo, data pubblicazione, numero battute, lingua.

Q6. Pulsanti e impostazioni grafiche coerenti con la proposta presente sul sito web

Q7. Barre di caricamento durante l'acquisizione di dati

2.1.3 Requisiti di vincolo (V)

V1. Un utente che non abbia effettuato l'accesso non può leggere, scaricare e votare l'opera

V2. La registrazione utente nell'applicazione avviene con solo Username, Mail e Password

V3. Nella ricerca avanzata, il parametro relativo alle lingue viene aggiornato automaticate all'accesso dell'utente recuperando le informazioni dal database

V4. A seconda della lingua del device, l'homepage caricherà informazioni distinte basate sulla lingua

V5. La sezione Leggi è divisa in tre categorie: Più recenti, Più lette, Migliori

V6. I risultati della ricerca vengono divisi in tre categorie: Più recenti, Più Lette, Migliori

2.1.4 Requisiti d'ambiente (A)

A1. Per interfacciarsi al Database si deveno utilizzare, per quanto possibile, framework e codice disponibili di TaleMotion

Per prendere una trattazione breve dei Requisiti di Ambiente, si riporta di seguito il lavoro svolto nello studio e nell'approccio alla risoluzione dei requisiti sopra indicati:

2.1.4.1 Framework utilizzati

L'analisi di framework e codici già esistenti all'interno del sito web per acquisire le informazioni necessarie alla prosecuzione della stessa hanno portato ad individuare solo due moduli utili allo sviluppo dell'applicazione:

- download_ebook.php
- addComment.php

2.1.5 Requisiti di interfaccia (I)

I1. La Homepage presenta una Slide iniziale e due liste di opere suddivise in In primo piano e Le migliori opere della settimana

I2. Le icone utilizzate sono caratteri UNICODE del font Font Awesome[1]

I3. I colori utilizzati devono essere coerenti con le scelte grafiche presenti nel sito web

I4. Rating Bar presente sull'applicazione deve essere coerente con la Rating Bar presente nel sito web

I5. ActionBar dinamica che presenti il logo di TaleMotion e pulsanti necessari a seconda della sezione visitata

2.1.6 Requisiti Software (S)

Con "Requisiti software" si intendono i software messi a disposizione dell'azienda per l'analisi e lo studio dei moduli e delle funzionalità presenti e necessari all'evoluzione dell'applicazione; vengono inoltre aggiunti i software che sono risultati necessari alla progettazione della stessa.

S1. **Apache**[2]
Apache è un software free e open-source sviluppato sulla base del server NCSA a partire dal 1994; implementa un server Web secondo le specifiche del protocollo HTTP. Il suo compito è quello di accettare richieste http da un host (richiesta di un documento o, più in generale, di una risorsa) e recuperare i dati necessari salvati nella memoria del dispositivo per ritornarle al richiedente iniziale. È stato scelto questo software perché presenta numerosi vantaggi: è open-source, portabile (OS Linux, Windows, ecc...), presenta un'architettura modulare, ha un buon supporto ai protocolli (conformità con HTTP/1.1), è flessibile ed affidabile.

S2. **PHP**[3]
PHP acronimo ricorsivo[4] di PHP: Hypertext Preprocessor è un linguaggio di programmazione interpretato, principalmente usato per sviluppare applicazioni Web lato server ma anche per scrivere script a riga di comando o applicazioni standalone. Originariamente fu concepito per la programmazione Web ovvero per la realizzazione di pagine web dinamiche.
Nato nel 1994, la versione attuale è una riscrittura ed estensione dell'autore Rasmus Lerdorf assieme a Zeev Suraski e Andi Gutmans (ideatori di Zend, il core di Php) ed offre numerose posiblità, tra cui l'interazione con numerosi DBMS e con codice HTML.
Il funzionamento all'interno del meccanismo client/server è il seguente: il client richiede una risorsa al server il quale, in accordo con Php, costruisce dinamicamente la pagina da ritornare, secondo con le istruzioni contenute in un file con estensione

[1]Font Awesome: http://fortawesome.github.io/Font-Awesome/
[2]Sito Web: http://www.apache.org/
[3]Sito Web: http://php.net/
[4]L'acronimo ricorsivo è una combinazione di parole nella quale compare l'acronimo stesso

php. Il risultato dell'elaborazione viene quindi ritornato al client, che visualizzerà sul proprio browser una classica pagina Web statica.

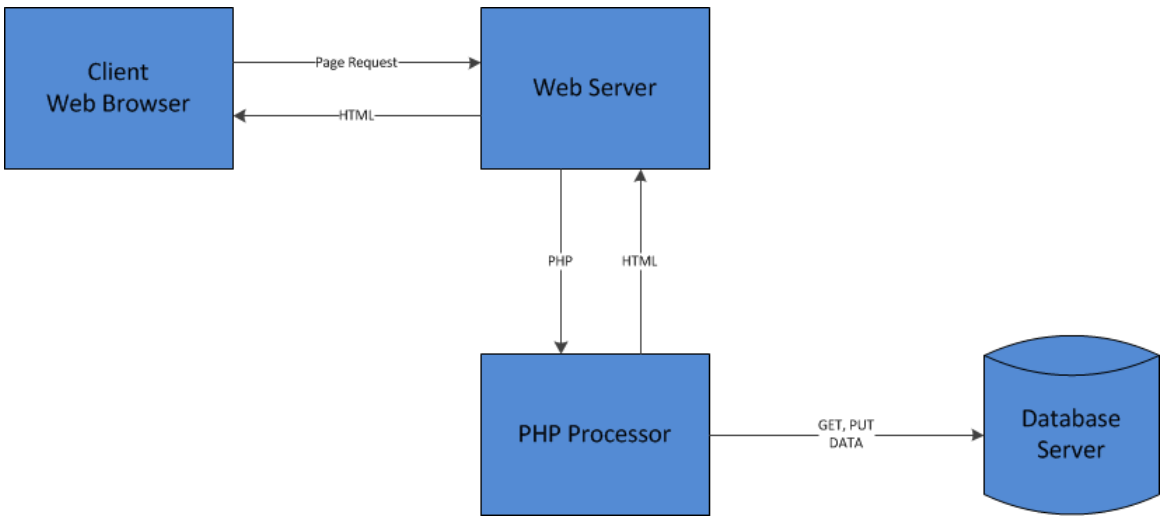


Figura 2.1: scambio di messaggi client/server/DBMS per la generazione di una pagina web dinamica

S3. **SQL**
SQL è un linguaggio di programmazione nato negli anni settanta per interrogare e gestire basi di dati. Per fare ciò si usano particolari costrutti di programmazione chiamati query.
Originariamente progettato come linguaggio di tipo dichiarativo, si è succesivamente evoluto con l'introduzione di costrutti procedurali, istruzioni per il controllo di flusso, tipi di dati definiti dall'utente e varie altre estensioni del linguaggio.
I comandi SQL utilizzati si dividono in:

- DDL (Data Definition Language): permette di creare e cancellare database o di modificarne la struttura.
- DML (Data Manipulation Language): permette di leggere, inserire, cancellare e modificare i dati.
- DCL (Data Control Language): permette di gestire gli utenti e i permessi.
- DMCL (Device Media Control Language): permette di controllare i supporti dove vengono memorizzati i dati.

Nella programmazione di pagine web, ad esempio, ci si appoggia spesso a DBMS, ovvero software o interfacce grafiche progettati per consentire la creazione, la manipolazione e l'interrogazione efficiente del database.

S4. **Navicat**[5]
Navicat è un tool grafico per gestire database MySQL, Oracle e PostGreSQL. In

[5]Sito Web: http://www.navicat.com/

particolare è possibile gestire efficacemente le connessioni a DBMS locali e remote, eseguire script SQL, esportare singole tabelle o interi DB, importare dati in formato cvs, xml, html o txt.

S5. **Android SDK**[6]
Android Software Development Kit (SDK) è un insieme di strumenti legato al sistema operativo Android. Android SDK include esempi di progetti con codice sorgente, strumenti per lo sviluppo, Debugger, un emulatore, relativa documentazione per API Android e le librerie necessarie per la costruzione di applicazioni Android. Il linguaggio di programmazione per la creazione di applicazioni è Java, il quale si appoggia a Dalvik, una macchina virtuale personalizzata appositamente per i sistemi embedded basati su sistema Linux.

Infine si vuole chiarire, in maniera sintetica, lo studio affrontato prima di poter iniziare lo sviluppo su piattaforma Android.
Punto cruciale è stato capire il funzionamento del Database utilizzato dall'azienda per il sito di TaleMotion: attraverso Navicat (Item 2.1.6.S4) si è appreso l'utilizzo e l'articolazione del Database interno: il passo è risultato basilare per sviluppare un'applicazione di questo genere in cui i dati risiedono all'esterno dell'applicazione stessa e vengono caricati in maniera dinamica dietro input dell'utente.
Per quanto riguarda lo sviluppo di interfaccia utente i requisiti richiesti dall'azienda prevedevano una coerenza fra la grafica presente sul sito web talemotion.com e l'applicazione che si sarebbe sviluppata. È risultato necessario capire gli stili e le formattazioni utilizzate dagli stessi per poter esaudire i requisiti richiesti. In sezione 3.4.3 verrà affrontato l'esempio di stile RatingBar.

[6]Sito Web: http://developer.android.com/sdk/index.html

Capitolo 3

Sviluppo

In questo capitolo viene affrontato lo sviluppo che ha portato all'evoluzione dell'applicazione di TaleMotion. Sono analizzati in maniera consistente funzionamenti e metodi di Android, dando un'analisi generica di come lo sviluppo sia proseguito. Sono inoltre introdotte porzione di codice PHP e codice JSON, motivando scelta e uso degli stessi.

3.1 Stabilire una connessione

Primo punto attuato per lo sviluppo dell'applicazione è stato quello di riuscire a connettere il device al sito di talemotion.com per poter accedere alle informazioni utili.
Android, fortunatamente, mette a disposizione diverse classi, con metodi annessi, che simulano in tutto e per tutto una chiamata POST[1] e GET[2] come farebbe un qualsiasi Client che deve connettersi ad un Server attraverso protocollo HTTP/1.1[3]. Di seguito vengono riportate le classi utilizzate massivamente per l'evoluzione dell'applicazione stessa.

1. DefaultHttpClient[4]
2. HttpPost[5]
3. HttpGet[6]
4. HttpResponse[7]
5. HttpEntity[8]

Queste poche classi, di cui viene descritta solo DefaultHttpClient (Item 1), sono bastate per permettere la connessione del device a un qualunque sito esterno, nella specifico a talemotion.com

[1] RFC 2616 Sezione 9.5
[2] RFC 2616 Sezione 9.3
[3] Sarebbe possibile anche usare il protocollo HTTP/1.0 ma non vi è alcun vantaggio nell'applicare questa procedura
[4] API: http://developer.android.com/reference/org/apache/http/impl/client/DefaultHttpClient.html
[5] API: http://developer.android.com/reference/org/apache/http/client/methods/HttpPost.html
[6] API: http://developer.android.com/reference/org/apache/http/client/methods/HttpGet.html
[7] API: http://developer.android.com/reference/org/apache/http/HttpResponse.html
[8] API: http://developer.android.com/reference/org/apache/http/HttpEntity.html

3.1.0.1 DefaultHttpClient (Item 1)

DefaultHttpClient è la classe che istanzia un Client per permettere una connessione attraverso il protocollo HTTP/1.1
Il costruttore standard, privo di parametri, è stato adoperato in tutta l'applicazione attraverso l'utilizzo del Design Pattern Singleton[9].
Di seguito viene riportato il codice che implementa il seguente Design Pattern.

Listing 3.1: SingletonHttpClient.java

```
package com.talemotionframework.singleton;

import org.apache.http.impl.client.DefaultHttpClient;
public class SingletonHttpClient {
        private static DefaultHttpClient mClient;
        private SingletonHttpClient() {};
        public static DefaultHttpClient getInstance() {
            if(mClient==null)
                mClient = new DefaultHttpClient();
            return mClient;
        }
}
```

Viene, successivamente, utilizzato il metodo prima definito per istanziare un DefaultHttpClient

Listing 3.2: Esempio di Utilizzo di Singleton (Listing 3.1)

```
DefaultHttpClient mClient = SingletonHttpClient.getInstance();
```

3.1.0.2 Perchè utilizzare il Design Pattern Singleton?

Il Design Pattern Singleton ha lo scopo di garantire che di una determinata classe venga creata una e una sola istanza.
Applicata a un DefaultHttpClient garantisce una continuità nella sessione HTTP.
Con questo si intende anche il passaggio e il salvataggio di Cookies in caso vengano trasmessi dal server.
Data l'implementazione del sito web www.talemotion.com, il download di un'opera può avvenire solo con la presenza di un cookies che accerti l'accesso fatto dall'utente. Non si renderebbe necessario per singole richieste a pagine web, ma dato l'ambiente in cui è calato, è servito a raggiungere lo scopo prestabilito.

[9]Il Singleton è un design pattern creazionale che ha lo scopo di garantire che di una determinata classe venga creata una e una sola istanza

3.2 AsyncTask

Una volta riuscita a stabile la connessione a un server attraverso il metodo spiegato in Sezione 3.1 si è posti il dubbio di come controllare e gestire l'aggiornamento dell'interfaccia utente per rendere l'applicazione di utilizzo semplice e immediato.
Per la gestione del UI Thread[10] è stata messa a disposizione al classe astratta[11] AsyncTask.

3.2.1 Descrizione

AsyncTask permette di eseguire in background operazioni e pubblicare i risultati sul UI-Thread senza aver bisogno di utilizzare e manipolare thread e/o handlers, è designata ad essere una classe d'aiuto per Thread e Handler e non costituisce in alcun modo un framework generico per il threading e dovrebbe essere usata idealmente per operazioni di breve periodo (pochi secondi).
Un task asincrono è definito da una computazione che viene eseguita su un thread in background e il cui risultato viene pubblicato sul UIThread. Un task asincrono è definito da quattro passi onPreExecute, doInBackground, onProgressUpdate e onPostExecute e da tre tipi di dati generici: Params, Progress e Results.

3.2.1.1 Tipi Generici

Params il tipo di dato che viene passato al task prima di essere eseguito.

Progress il tipo di dato che viene pubblicato durante lo svolgimento della computazione in background

Result il tipo di dato risultato alla fine della computazione in background

3.2.1.2 I quattro passi

onPreExecute viene invocato sul UI Thread prima che il task venga eseguito. Questo passo viene normalmente chiamato per disporre il task, ad esempio mostrando una progress bar all'utente.

doInBackground(Params...) viene invocato dal thread in background subito dopo che onPreExecute() termina. Questo passo è usato per eseguire il compito assegnato all'AsyncTask impiegando anche un tempo relativamente lungo. In questo passo si può utilizzare anche il metodo publishProgress(Progress...) per rendere disponibili i valori che si stanno acquisendo. Questi valori sono pubblicati sul UIThread nel passo onProgressUpdate()

[10] L'UIThread è il thread principale dell'esecuzione dell'applicazione. È dove la maggior parte del codice dell'applicazione viene eseguito. Tutti i componenti dell'applicazione (Activity, Service, ContentProvider, BroadcastReceive) sono creati in questo thread e qualsiasi chiamate a questi ultimi sono eseguite in questo thread. http://developer.android.com/guide/components/processes-and-threads.html

[11] Si definisce classe astratta una classe che definisce una interfaccia senza implementarla completamente.

onProgressUpdate(Progress...) viene invocato dal UI Thread dopo che viene chiamato il metodo publishProgress(). Il tempo di esecuzione di questo passo è indefinito. Questo metodo è usato solitamente per mostrare all'utente una sorta di progresso della computazione che si sta eseguendo in background.

onPostExecute(Result) viene invocato dal UI Thread dopo che il calcolo in background è terminato. Il risultato della computazione prima eseguita viene passata come parametro in questo passaggio.

3.2.2 Utilizzo

AsyncTask, essendo una classe astratta, deve essere implementato attraverso una sottoclasse che estenda quest'ultima.
La sottoclasse deve sovrascrivere almeno il metodo doInBackground(Params...), e, come molto spesso accade, viene sovrascritto anche il metodo onPostExecute(Result).
Una volta creata la sottoclasse, per eseguire il task bisogna invocare il metodo *execute(Params... params)*.
Il metodo *execute(Params... params)* esegue il task con i parametri specificati. Il metodo ha come parametro di ritorno sè stesso, così che il chiamante ha una referenza a sè.

3.2.3 Esempio di Implementazione

In Appendice A (Listing 6.1, 6.2) viene mostrato un esempio di implementazione della classe AsyncTask.
L'esempio è l'implementazione del Download eBook di un'opera nel formato scelto dall'utente.
Di seguito è presente una descrizione dei metodi e delle classi che appaiono nel codice citato.
A conclusione della sezione verranno mostrate immagini che rispecchiano l'utilizzo sull'emulatore.

3.2.3.1 Descrizione del codice

onPreExecute()

lockScreenOrientation()

Il metodo statico, non visualizzato in codice, serve a impedire al device la rotazione dello schermo. Questo metodo è stato implementato con lo scopo di prevenire arresti improvvisi dell'applicazione causati dalla rotazione schermo. Android infatti, passando da modalità portrait a landscape e viceversa, distrugge l'activity corrente per ricrearne una nuova. Il problema che può occorrere riguarda proprio l'AsyncTask e il controllo dell'interfaccia grafica; se infatti, durante l'esecuzione del task in background, si andasse a ruotare lo schermo, non solo verrebbe creata una nuova istanza della classe DownloadBookASyncTask (in caso di nuovo download), ma inoltre la precedente istanza, ancora attiva,

avrebbe riferimenti a oggetti ormai nulli per ciò che riguarda l'interfaccia grafica, generando quindi l'eccezione *NullPointerException()*. Questo metodo è stato introdotto come workaround all'uso di handler. Bisogna ricordare che in caso di rotazione schermo, l'Activity passa attraverso lo stato *onPause*[12] nel quale si potrebbe, e si deve, intervenire. In effetti nell'implementazione nel metodo *onPause()* dell'Activity, viene chiamato il metodo che cancella l'esecuzione in background della classe, in modo da prevenire il generarsi di arresti anomali.

pDialog
Viene creata dunque una Dialog (*new Dialog(mContext)*), specificandone un layout personalizzato (*setContentView()*), assegnandole un titolo (*setTitle('Downloading')*), rendendola non cancellabile (*setCancelable(false)*, *setCanceledOnTouchOutside(false)*) e infine mostrandola all'utente (*show()*).

doInBackground()

mClient
Viene creato un nuovo DefaultHttpClient attraverso l'istanza Singleton (Listing 3.1)

ResourceString
ResourceString è un file in cui sono presenti tutti valori utili per l'applicazione. In questo caso vengono utilizzati per avere l'URL a cui connettersi per eseguire la chiamata Post e Get.

HttpPost, HttpGet
Classi utilizzate per la gestione di POST e GET (Ref. 2, 3)

BasicNameValuePair [13]
Una semplice classe che permette l'incapsulamento di coppie attributo/valore. Questa classe è conforme alla grammatica generica presente in Sezione 2.2 e Sezione 2.3 dell'RFC2616[14]. Come si nota dal codice, vengono assegnati all'attributo username e all'attributo password parametri ricevuti dal metodo execute.

HttpResponse
Semplice classe che emula una Response HTTP[15].

HttpEntity
Della HttpResponse siamo interessati all'Entity[16], cioè al contenuto privo di Headers. Otteniamo l'entity applicando il metodo *getEntity()* all'HttpResponse.

[12]Ogni Activity ha un ciclo di vita, durante il quale essa passa attraverso vari stati. Ad ogni stato viene associato un metodo per gestirne il passaggio. Il metodo onPause() viene invocato quando un'altra Activity passa in foreground. http://developer.android.com/reference/android/app/Activity.html
[13]API: http://developer.android.com/reference/org/apache/http/message/BasicNameValuePair.html
[14]RFC HTTP\1.1: http://tools.ietf.org/html/rfc2616
[15]RFC 2616 Sec 6
[16]RFC 2616 Sec 7

Da questo siamo in grado di ricevere lo stream di dati che compongo l'eBook scaricato.

File,InputStream,FileOutputStream
Sono tutte classi che permettono la gestione di File, la creazione di esso, la scrittura e l'analisi.
Dal codice si legge che il contenuto dell'Entity viene letto e scritto in un buffer, una volta conclusa la lettura il buffer viene riciclato e il file viene chiuso ottenendo così il nostro eBook nel formato desiderato.

return new String[ ]
Il metodo doInBackground è stato dichiarato con la presenza di un array di stringhe come parametro di uscita. Nel caso utilizzato servirà al metodo onPostExecute() per avere il fullpath del file che si vuole aprire con relativa estensione.

onPostExecute()

unlockScreenOrientation()
Il metodo, non mostrato in codice, concede nuovamente al device la rotazione dello schermo, in quanto, avendo terminato il task in background dell'AsyncTask garantisce la prosecuzione della computazione senza rischio di incorrere in alcuna eccezione.

pDialog.dismiss()
Terminato il processo di Download dell'eBook, la Dialog prima mostrata all'utente ora viene cancellata.

startActivity
Metodo nativo di Android che permette di avviare una nuova Activity. Necessita come parametro implicito un Intent[17] passatogli attraverso *Utils.openFile*. Nel caso specifico l'Intent che viene lanciato riguarda l'apertura dell'eBook nel formato desiderato. Viene gestito il caso in cui non sia presente nessuna applicazione che possa aprire il file desiderato rimandando al Play Store di Android per il Download di un'applicazione esterna che possa gestire la lettura del file stesso.

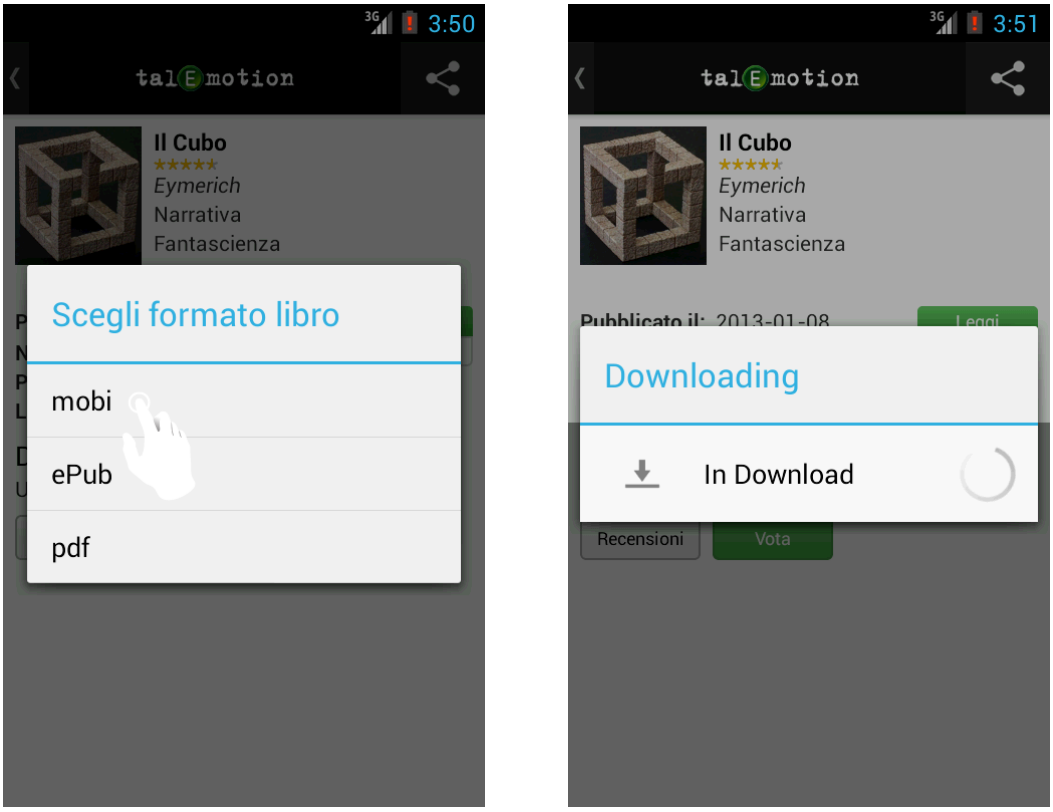
Utils
Utils è una classe creata per la gestione di metodi utili all'applicazione. Nel caso descritto viene utilizzato il metodo *openFile(String, String)* che restituisce un Intent il cui Action è impostato a ACTION_VIEW[18].

3.2.3.2 Immagini esemplificative

Come precedentemente specificato, di seguito due immagini esemplificative di ciò che accade su un device.

[17] L'Intent è la descrizione astratta di una operazione che deve essere eseguita. http://developer.android.com/reference/android/content/Intent.html

[18] L'Action di un Intent è l'azione generica che deve compiere l'Intent. ACTION_VIEW indica che l'intent dovrà mostrare dei dati all'utente



(a) Click sull'estensione scelta (b) Dialog con Progress Bar

Figura 3.1: Emulazione del codice su un virtual device

3.3 Interfacciarsi al Database

Fino ad ora si è parlato di come il device si possa connettere alla rete (Sezione 3.1) e come gestire l'UI Thread tramite la classe AsyncTask (Sezione 3.2).
Ci si pone ora il dubbio di come poter interrogare il Database per ottenere le informazioni necessarie al corretto funzionamento dell'applicazione.

3.3.1 Approccio

Lo studio, oltre alla richiesta dell'azienda di utilizzare per quanto possibile framework esistenti, ha portato alla seguente soluzione:

1. Client (Android Device) chiama una pagina PHP
2. Pagina PHP interroga il database e processa i dati
3. PHP crea la pagina statica contenente i dati in formato JSON
4. Client (Android Devide) interpreta il formato JSON e aggiorna l'applicazione

La soluzione appena descritta è sicuramente la più diffusa e utilizzata in ambito Database, infatti Android sconsiglia l'utilizzo di librerie per connetteresi a Database esterni.

3.3.2 JSON

3.3.2.1 Descrizione

JSON[19] (**J**ava**S**cript **O**bject **N**otation) è un semplice formato per lo scambio di dati. Per gli utenti è facile da leggere e scrivere, mentre per le macchine risulta facile da generare e analizzarne la sintassi.

JSON è basato su due strutture:

- Un insieme di coppie nome/valore. In diversi linguaggi, questo è realizzato come un oggetto, un record, uno struct, un dizionario, una tabella hash, un elenco di chiavi o un array associativo.
- Un elenco ordinato di valori. Nella maggior parte dei linguaggi questo si realizza con un array, un vettore, un elenco o una sequenza.

Queste sono strutture di dati universali. Virtualmente tutti i linguaggi di programmazione moderni li supportano in entrambe le forme. È sensato che un formato di dati che è interscambiabile con linguaggi di programmazione debba essere basato su queste strutture.

[19] Sito web: http://www.json.org/

In JSON si possono assumere queste forme:

Un *Oggetto* è una serie non ordinata di nomi/valori. Un oggetto inizia con { (parentesi graffa sinistra) e finisce con } (parentesi graffa destra). Ogni nome è seguito da **:** (due punti) e la coppia di nome/valore sono separata da **,** (virgola).

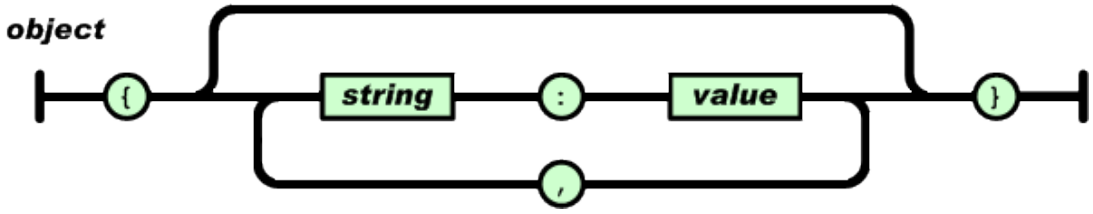


Figura 3.2: Struttura di un Oggetto JSON

Un *Array* è una raccolta ordinata di valori. Un array comincia con [(parentesi quadra sinistra) e finisce con] (parentesi quadra destra). I valori sono separati da **,** (virgola).

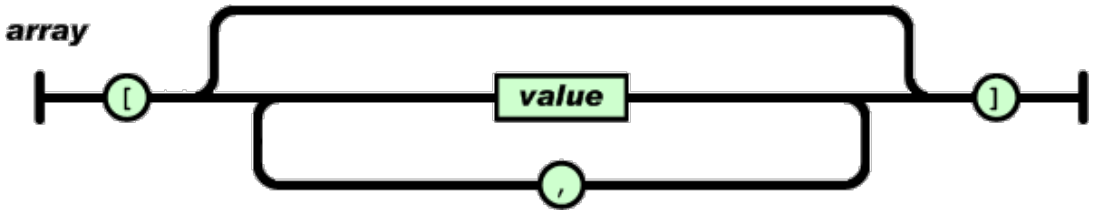


Figura 3.3: Struttura di un Array JSON

Un *Valore* può essere una stringa tra virgolette, o un numero, o vero o falso o nullo, o un oggetto o un array. Queste strutture possono essere annidate.

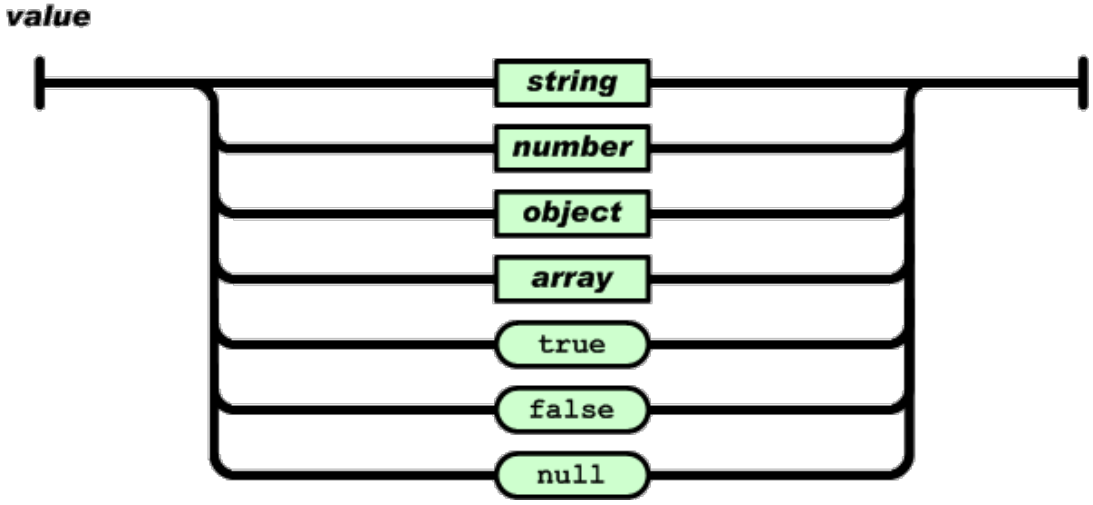


Figura 3.4: Struttura di un Valore JSON

3.3.3 Creazione pagina PHP

Capito dunque l'approccio da seguire, sono state create diverse pagine PHP necessarie alla prosecuzione dell'evoluzione dell'applicazione.
Ogni pagina si comporta in modo simile:

1. Si stabilisce connessione al Database
2. Si costruisce in maniera dinamica la Query da eseguire
3. I dati ritornati vengono convertiti in JSON e stampati a schermo

Viene di seguito mostrato il codice utilizzato nella pagina *Recensioni.php*

Listing 3.3: Recensioni.php, l'introduzione dei simboli "?" è dovuta alla riservatezza del codice e del Database dell'azienda

```
<?php
$libdir = "../PhpLib/";
require_once($libdir."prepend.php");
require_once($libdir."../config.inc.php");
$id = GetParameter("id_book","");
$myQuery = "SELECT A.?, date(A.?) as datacommento,".
    " A.? as titolocommento,".
    " A.? as commento, B.? as slugute,".
    " C.? as scorecommento".
    " FROM ? as A".
    " join ? as B ON B.?=A.?".
    " join ? as C".
    " ON C.?=B.?".
    " WHERE A.?=1 and A.?=".$id.
    " GROUP BY C.? ORDER BY A.? desc;";
$rows = array();
$result = mysql_query($myQuery);
while($r = mysql_fetch_assoc($result)) {
    $rows[] = $r;
}
echo json_encode($rows);
?>
```

Nel codice sopra citato (Listing 3.3) si vede l'utilizzo di alcune funzioni create dall'azienda.
La funzione *GetParameter("","");* legge dagli array di GET e POST il valore dei parametri passati (se esistono).
I *require_once* presenti a inizio codice richiamano i file necessari che permettono di stabilire automaticamente la connessione al Database.
Come si nota, un qualsiasi Client che chiami questa pagina dovrà introdurre l'attributo "id_book" specificando l'id del libro di cui si vogliono avere le recensioni.
Funzione di estrema utlilità è stata *json_encode*[20], funzione nativa di PHP introdotta con PHP 5.0 la quale ritorna una stringa in formato JSON del parametro passato alla

[20]Manuale: http://www.php.net/manual/it/function.json-encode.php

funzione.
Per mostrare un esempio chiediamo alla pagina le recensioni dell'opera con id_book=86
Il risultato è mostrato in Appendice A (Listing 6.3).

3.3.4 Android

Da lato Android esistono classi e funzioni di interpretazione di oggetti JSON, da qui si individua la funzionalità dell'approccio seguito.
Le classi che permettono la codifica e decodifica di oggetti JSON sono le seguenti:

JSONArray [21]
: Una densa sequenza di valori indicizzati. I valori possono essere un qualsiasi numero dei seguenti tipi di dato: String, Boolean, Integers, Longs, Doubles, null, NULL, JSONObject o JSONArray stessi.
Si presenta una differenza fra *null* e *NULL*. La prima è la referenza null di *Java* mentre la seconda è il valore sentinella NULL proprio di *JSON*.

JSONObject [22]
: Un insieme modificabile di mappa nome/valore. Il valore può essere uno qualsiasi dei seguenti tipi di dato: String, Boolean, Integers, Longs, Doubles, NULL, JSONObject o JSONArray.

Le classi appena illustrate con i metodi relativi riescono a ricoprire le funzioni cercate per l'acquisizione di dati. Si veda l'utilizzo di esse nei relativi codici in Appendice A. Vengono mostrate due classi: 1) **LoadRecensioni** (Appendice A - Listing 6.4) classe che estende AsyncTask (Sezione 3.2) per gestire l'UI Thread, oltre che l'acquisizione dei dati. 2) **JSONParser** (Appendice A - Listing 6.5) classe costruita per l'esecuzione POST o GET HTTP\1.1 connettendosi a un URL, convertendo i dati ritornati in un JSONArray.
.

[21] API: http://developer.android.com/reference/org/json/JSONArray.html
[22] API: http://developer.android.com/reference/org/json/JSONObject.html

3.3.4.1 Emulazione su Device

Viene mostrata di seguito il risultato della richiesta delle recensioni dell'opera con id=86. Come si vede i dati sono coerenti con quelli mostrati in Appendice A - Listing 6.3.

Figura 3.5: Risultato su Emulatore Android alla richiesta di recensioni dell'opera con id=86

3.4 Interfaccia Utente

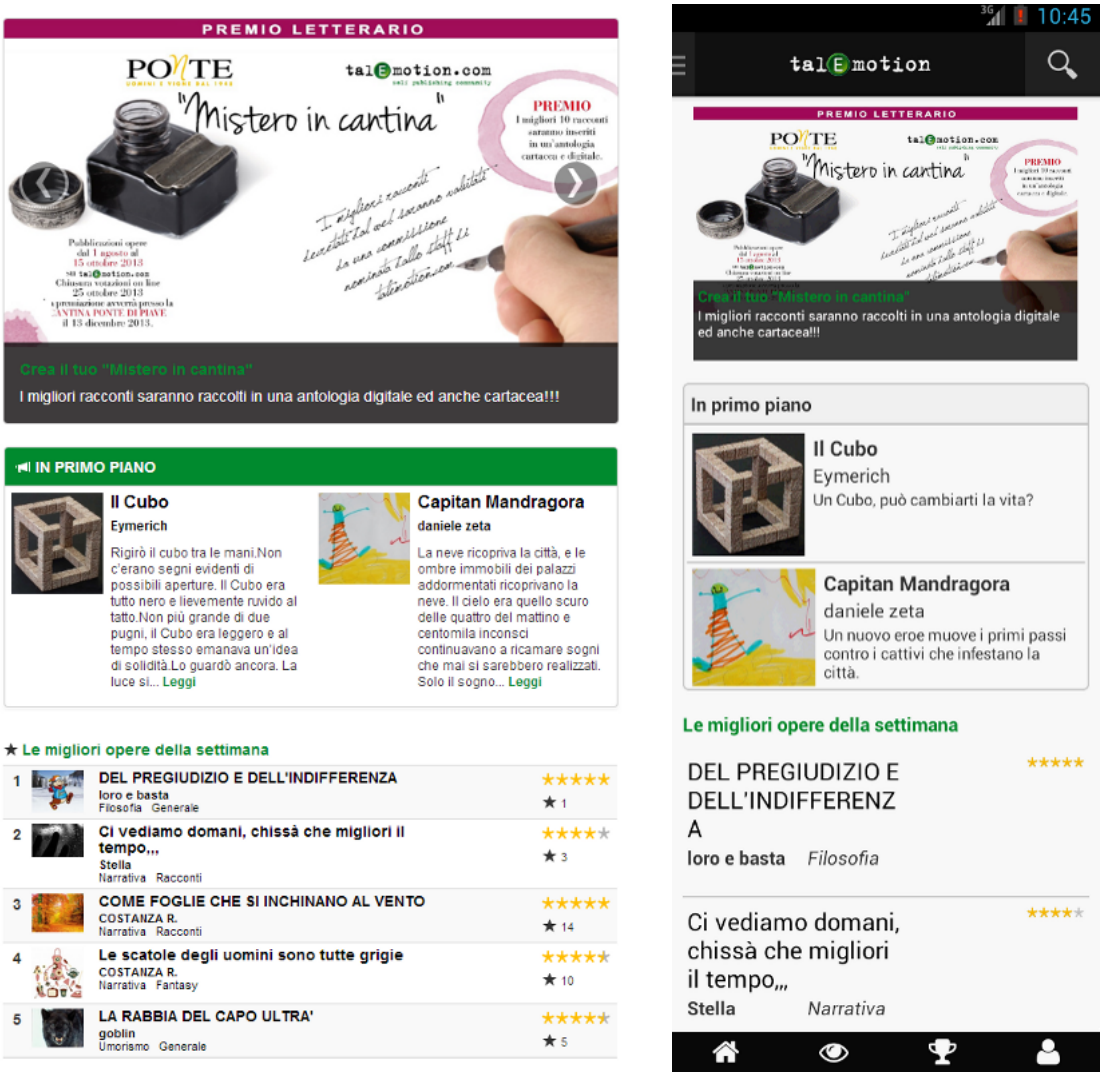
3.4.1 Introduzione

Nello sviluppo di un'applicazione per smartphone l'implementazione di una GUI[23] ben disegnata e di facile utilizzo deve essere considerato un aspetto fondamentale e imprescindibile.
L'importanza dell'interfaccia grafica viene molto spesso sottovalutata durante la realizzazione di un nuovo software: in ogni settore, esistono programmi molto potenti e sofisticati, ma dotati di un'interfaccia poco intuitiva, anonima e di scarsa qualità. La conseguenza immediata è che il programma diventa difficile e poco piacevole da usare.
Un'applicazione ben sviluppata ma dalla scarsa e insoddisfacente interfaccia può far fallire il prodotto stesso, mentre un prodotto non così ben ottimizzato ma con un'interfaccia pulita e user friendly può diventare di successo.
Nonostante sia stata lasciata come ultimo passo nella realizzazione del progetto e come ultima sezione della tesi, si deve sempre ricordare che l'interfaccia utente può decidere - in positivo o in negativo - le sorti del software al quale si applica.
Durante lo sviluppo le decisioni grafiche sono state analizzate per ultime, non perché all'interfaccia sia stata attribuita minor importanza rispetto ad altri elementi, ma solo a causa di contingenze temporali e di sviluppo che hanno reso impossibile occuparsene prima.

[23] Graphical user interface

3.4.2 Interventi Grafici

L'analisi e lo sviluppo grafico ha portato a decisioni grafiche coerenti con le proposte presenti sul sitoweb di www.talemotion.com
Si analizzi l'Homepage del sito con l'Homepage[24] dell'applicazione.



(a) Homepage www.talemotion.com (b) Prima Activity Applicazione

Figura 3.6: Homepage messe a confronto

[24]In un'applicazione non si può parlare di homepage quanto piuttosto di una prima View (schermata), ma la trattazione è allo stesso livello di una comune Homepage di sito web

La realizzazione dell'Activity di apertura dell'applicazione riesce ad essere il più coerente possibile con ciò che appare in homepage sul sito web.
Bisogna specificare alcuni dettagli di relativa importanza: l'utente finale che accede al sito web ha, solitamente, uno schermo di navigazione ampio data la grandezza del calcolatore che sta usando, di conseguenza le informazioni che possono apparire su schermo sono, e possono essere, in numero maggiore; differentemente, le applicazione per smartphone, ma ormai anche su tablet, devono essere trattate in maniera diversa proprio perché si tratta di hardware differenti: infatti le informazioni che vengono date all'utente devono essere in numero molto minore, quelle essenziali all'usufruibilità dell'applicazione stessa. Filosoficamente parlando, in un interfaccia web ci si può permettere di avere un approccio sincronico, in cui il tempo viene considerato un parametro fisso: le informazioni possono essere visualizzate in maniera completa, non preoccupandosi di dover dare un ordine cronologico a queste ultime data appunto la possibilità di visualizzarle interamente. In maniera del tutto diversa, su un device smartphone, a piccoli schermi, la scelta di un approccio diacronico risulta essenziale, scelta dovuta all'impossibilità di avere tutte le informazioni desiderate a schermo, ma collocate in ambiti e tempi diversi, obbligando l'input dell'utente per la visualizzazione dei dati non mostrati. Risulta dunque necessaria una gerarchia delle informazioni e dei dati.
A dimostrazione di ciò si noti in Figura 3.7 come appare l'homepage su un vero device. Le informazioni e i dati mostrati "direttamente" all'utente finale sono troncate e non vengono mostrate per intero, obbligando l'utente ad uno scroll verso il basso per poter raggiungere quest'ultime.
Bisogna dunque decidere quali sono i primi dati, le informazioni necessarie e più importanti da visualizzare senza obbligare l'utente a interagire con l'applicazione. I committenti dell'applicazione hanno optato per avere in prima pagina una slide e la lista delle opere scelte, questo ancora una volta per rispecchiare e essere coerenti con ciò che il sito web propone.

Figura 3.7: Applicazione su Device Samsung Galaxy S3

3.4.3 Un caso specifico: RatingBar

Una Rating Bar è una barra che permette di visualizzare un voto sottoforma di un'interfaccia grafica solitamente attraverso delle stelle.
L'analisi della Rating Bar è ideale per mettere in luce le ampie possibilità che Android ha a livello di personalizzazione grafica e la relativà semplicità del suo utilizzo; inoltre è stata specifica importante sull'interfaccia grafica richiesta dal cliente, per cui la sua trattazione è motivo di grande interesse per questo studio.

3.4.3.1 RatingBar su Android

La RatingBar è un'estensione della SeekBar e della ProgressBar che mostra un voto attraverso stelle. L'utente può toccare/trascinare o usare i tasti direzionali per impostare il voto. È possibile avere una RatingBar che sia di sola lettura, mostrando all'utente il voto ma impedendone la modifica.

3.4.3.2 Obiettivo

L'obiettivo della personalizzazione è di rendere coerente l'interfaccia utente alle scelte presenti sul sito web. Lo scopo è reso visibile dalle immagini sottostanti: la prima mostra la RatingBar di Default del sistema operativo Android con versione 4.0 superiore (Figura 3.8a), la seconda invece la Rating Bar prodotta dopo la personalizzazione (Figura 3.8b).

(a) RatingBar Default Android 4.0

(b) RatingBar personalizzata stile Talemotion

Figura 3.8: Stili RatingBar

3.4.3.3 Personalizzazione

Android è dotato di un supporto styling per la modifica della visualizzazione di determinati elementi grafici.
In poche parole, gli stili forniscono metodi per alterare l'interfaccia della grafica e delle componenti android attraverso i file XML[25] mantenendo il codice e le funzionalità inalterate.
I dati dell'UI vengono memorizzati in file XML e all'atto della compilazione essi divengono risorse poi usufruibili all'interno del codice stesso. Impadronitosi dei concetti di

[25]XML (sigla di eXtensible Markup Language) è un linguaggio di markup, ovvero un linguaggio marcatore per documenti contenenti informazioni strutturate

XML è stato necessario l'utilizzo del file *styles.xml*[26], file nativo di Android che definisce il formato e la visualizzazzione dell'UI che risiede all'interno della cartella *values*.

Listing 3.4: style.xml

```
<style name="RatingBarBigStyle" parent="android:Widget.RatingBar">
    <item name="android:progressDrawable">@drawable/ratingbarbig</item>
    <item name="android:minHeight">30dip</item>
    <item name="android:maxHeight">40dip</item>
</style>
```

Il codice appena illustrato (Listing 3.4) crea un nuovo stile personalizzato chiamato RatinBarBigStyle che estende lo stile Widget.RatingBar, impostando l'altezza minima a 30dip e quella massima ai 40dip, e selezionando la sua progressDrawable[27] a **ratinbarbig** (Risorsa presente nella cartella *drawable*).
Viene fornito di seguito anche il codice presente all'interno della risorsa *ratingbarbig* (Listing 3.5).

Listing 3.5: ratingbarbig.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:id="@+android:id/background"
    android:drawable="@drawable/ratingbarbig_full_empty" />
    <item android:id="@+android:id/secondaryProgress"
    android:drawable="@drawable/ratingbarbig_full_empty" />
    <item android:id="@+android:id/progress"
    android:drawable="@drawable/ratingbarbig_full_fill" />
</layer-list>
```

In pratica, viene fornita una lista di Drawable usati per il background (Nessuna stella – *ratingbarbig_full_empty*) e per l'avanzamento (Stella selezionata – *ratingbarbig_full_fill*). Questi Drawable sono selettori nei quali viene redatta una lista per le immagini da usare per i diversi stati nella selezione della RatingBar. Di seguito viene riportato il file *ratingbarbig_full_fill.xml* (Listing 3.6).

Listing 3.6: ratingbarbig_full_fill.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
    <item android:state_pressed="true"
    android:state_window_focused="true"
    android:drawable="@drawable/star_on_big" />

    <item android:state_focused="true"
    android:state_window_focused="true"
    android:drawable="@drawable/star_on_big" />
```

[26] API: http://developer.android.com/guide/topics/resources/style-resource.html

[27] Si ricordi che la RatingBar è un'estensione di una ProgressBar di cui ogni "stella" nella RatingBar è praticamente solo un'altra tacca nella ProgressBar

```
11      <item android:state_selected="true"
        android:state_window_focused="true"
13      android:drawable="@drawable/star_on_big" />

15      <item android:drawable="@drawable/star_on_big" />

17  </selector>
```

Viene utilizzata una sola immagine per tutti e quatro gli stati, ma come si può vedere dal *selector* ci sono quattro possibili stati diversi, di conseguenza si può applicare un immagine diversa per ogni stato possibile. Infine @drawable/-star_on_big è l'effettiva immagine della stella qui mostrata (Fig. 3.9).

Figura 3.9: star_on_big.png

La cosa interessante è che Android riempirà in maniera automatica la RatingBar con la parte necesseria al voto che questo implichi una stella piena, una stella vuota o una mezza stella.
L'ultimo codice che viene mostrato riguarda l'applicazione della custom RatingBar a una RatingBar introdotta in una schermata.

Listing 3.7: Come applicare lo stile

```
1  <RatingBar android:id="@+id/my_rating_bar"
               ...
3              style="@style/RatingBarBigStyle" />
```

Capitolo 4

Conclusioni

In questa tesi sono state presentate le attività svolte per lo sviluppo dell'applicazione su piattaforma Android del sito www.talemotion.com.
Il percorso effettuato ha avuto una ripartizione temporale e procedurale, in quanto lo sviluppo di un'applicazione procede secondo passi che, con l'avanzare dell'evoluzione stessa, vengono svolti e ampliati per raggiungere l'obiettivo finale.
In sintesi, dal problema generale - ottenere un'applicazione fruibile - sono stati ricavati dei sotto problemi, qui elecanti, svolti e sviluppati in maniera gerarchica per il conseguimento e la risoluzione del problema padre:

1. Analisi del prodotto TaleMotion.com, con relativo studio di framework, database e funzionamento generale e specifico dello stesso.

2. Analisi e studio degli strumenti Android messi a disposizione per la realizzazione dell'applicazione.

3. Analisi dell'approccio da seguire per l'evoluzione dell'applicazione.

4. Sviluppo pagine PHP.

5. Sviluppo tecnico dell'applicazione: il main core di TaleMotion

6. Analisi grafiche e studio di fattibilità.

7. Implementazione grafiche sull'applicazione TaleMotion.

Di seguito viene mostrata, attraverso un grafico a torta (Fig. 4.1) una stima delle risorse impiegate in termini di tempo per la realizzazione dei punti precedenti.

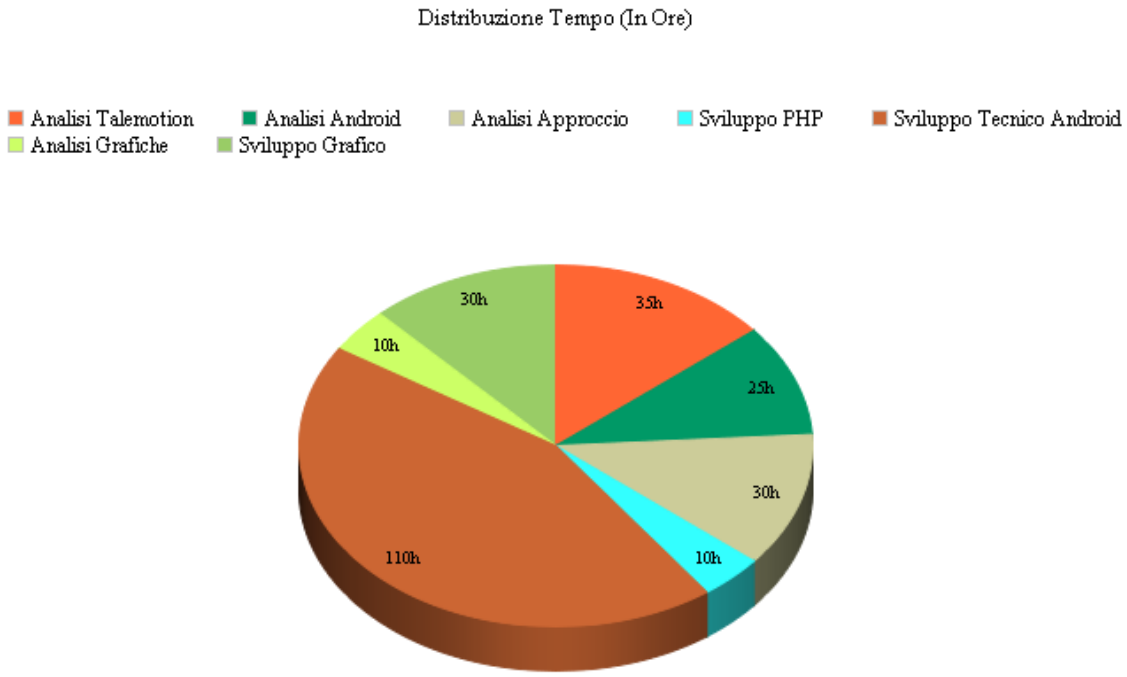


Figura 4.1: Grafico a torta con la distribuzione di ore per il lavoro svolto

Lo sviluppo dell'applicazione ha riscontrato esiti positivi così come dimostra la recensione, proposta di seguito, di *Andrea Grasso*, presidente ARS Network, e di *Roberto Quaglio*, socio ARS Network, riguardo il lavoro svolto:

> "*L'applicazione rispetta in pieno le caratteristiche tipiche di una App che deve replicare le funzionalità salienti di un sito per altro molto articolato, quando si tratta di un Social Network, quindi di una piattaforma che preveda non solo contenuti, ma anche contatti tra utenti, dinamiche relazionali virtuali, responsività.*
> *Il prodotto risulta molto soddisfacente per livello di facilità d'uso, grafica coerente con il sito, timing di risposta. Le principali funzionalità di base sono replicate nella App.*
> *Il tempo limitato di stage lascia aperte le possibilità per il completamento di alcune parti riguardanti la sezione "concorsi", "posta" e "impostazioni".*"

L'applicazione proposta finora è una prima versione, pronta ad essere pubblicata sul Play Store Android.
Nelle prossime versioni si implementarà il multilingua (in parte già presente): fortunatamente Android mette a disposizione un modo per risolvere questo problema, dunque sarà una delle prime feature che verrà implementata.

Dovranno essere implementate due nuove sezioni - “concorsi” e i “profili utenti” - così come il modulo “impostazioni” ancora mancante.
Caratteristica importante che dovrà essere implementata riguarda la parte “Social” del sito www.talemotion.com: gli utenti, sul sito web, possono inviare e ricevere messaggi; questa peculiarità non è stata ancora analizzata, ma è di molto interesse per i committenti.
Infine si vorrà implementare un sistema di notifiche che avvisi l’utente sulla presenza di nuovi messaggi o di commenti da approvare.

Capitolo 5

Bibliografia

- Carli, M. (2011). 1. *Android 3: guida per lo sviluppatore : [scoprire la soluzione di Google per dispositivi mobili, realizzare applicazioni per smartphone e tablet]* (pp. 3,4). Milano: Apogeo.
- Documentazione Android. (n.d.). *Android Development.* Consultato 30 Agosto, 2013, https://developer.android.com/reference/packages.html
- Documentazione JSON. *JSON.* Consultato 30 Agosto, 2013, http://www.json.org/
- Documentazione PHP. *PHP: Hypertext Preprocessor.* Consultato 30 Agosto, 2013, from http://php.net/docs.php
- Sito EITEAM. *EITEAM scs.* Retrieved August 30, 2013, http://www.eiteam.it
- Sito Talemotion. *Self publishing community - Talemotion - Self Publishing Community.* Consultato 30 Agosto, 2013, http://www.talemotion.com/

Capitolo 6

Appendice A

Listing 6.1: Chiamata al metodo execute della classe DownloadBookASyncTask

```
new DownloadBookASyncTask().execute(
    getSharedPreferences(ResourceString.PREFS_NAME,0)
        .getString(ResourceString.PREFS_USERNAME_CLIENT,""),
    getSharedPreferences(ResourceString.PREFS_NAME,0)
        .getString(ResourceString.PREFS_PASSWORD_CLIENT,""),
    ""+idbook,
    f.getPath(),
    myExt);
```

Listing 6.2: Classe DownloadBookASyncTask che estende AsyncTask

```
class DownloadBookASyncTask extends AsyncTask<String,String,String[]> {
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        lockScreenOrientation();
        pDialog=new Dialog(mContext);
        pDialog
        .setContentView(R.layout.activity_scheda_book_customdialog);
        pDialog.setTitle("Downloading");
        pDialog.setCancelable(false);
        pDialog.setCanceledOnTouchOutside(false);
        pDialog.show();
    }
    @Override
    protected String[] doInBackground(String... params) {
        DefaultHttpClient mClient = SingletonHttpClient.getInstance();
        HttpPost httpPost = new HttpPost
                        (ResourceString.HTTPREQUEST_LOGINACTIVITY);

        List <NameValuePair> nvps = new ArrayList <NameValuePair>();
        nvps.add(new BasicNameValuePair("username",params[0]));
        nvps.add(new BasicNameValuePair("password",params[1]));
```

```
24          try {
                httpPost
26              .setEntity(new UrlEncodedFormEntity(nvps, HTTP.UTF_8));
                mClient.execute(httpPost);
28          } catch (ClientProtocolException e) {
                e.printStackTrace();
30          } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
32          } catch (IOException e) {
                e.printStackTrace();
34          }
            String url = ResourceString.HTTPREQUEST_DOWNLOADBOOK +
36                  "?book=" + params[2] + "&type=" + params[4];
            HttpGet httpGet = new HttpGet(url);
38          try {
                HttpResponse response = mClient.execute(httpGet);
40              HttpEntity mEntity = response.getEntity();
                File f = new File(params[3]);
42              FileOutputStream fOut = new FileOutputStream(f);
                InputStream is=mEntity.getContent();
44              byte data[] = new byte[1024];
                int count;
46              while ((count = is.read(data)) != -1) {
                  fOut.write(data, 0, count);
48              }
                fOut.flush();
50              fOut.close();
                is.close();
52
            } catch (ClientProtocolException e) {
54              e.printStackTrace();
            } catch (IOException e) {
56              e.printStackTrace();
            }
58          return new String[] {params[3],params[4]};
        }
60      @Override
        protected void onPostExecute(String[] arg) {
62          unlockScreenOrientation();
            pDialog.dismiss();
64          try {
                startActivity(Utils.openFile(arg[0],arg[1]));
66          }
            catch(ActivityNotFoundException activityNotFoundException) {
68              Toast.makeText
                    (mContext, "Nessuna App trovata", Toast.LENGTH_LONG)
70                  .show();
                Intent intent = new Intent(Intent.ACTION_VIEW);
72              intent.setData
```

```
                    (Uri.parse("http://play.google.com/store/search?q="
74                  +arg[1]+"&c=apps"));
                startActivity(intent);
76          }
        }
78  }
```

Listing 6.3: Risultato della richiesta a Recensioni.php (Listing 3.3) con id_book=86


```
[
    {
        "id":"293",
        "datacommento":"2012-10-22",
        "titolocommento":"Piaciuto!",
        "commento":"Complimenti, sono rimasto davvero colpito!",
        "slugute":"Guy Fawkes",
        "scorecommento":"4.50"
    },
    {
        "id":"248",
        "datacommento":"2012-07-29",
        "titolocommento":"Geniale",
        "commento":"Ispidi istrionici istanti...inno alla \"i\"",
        "slugute":"rosa bellini","scorecommento":"4.00"
    },
    {
        "id":"247",
        "datacommento":"2012-07-29",
        "titolocommento":"",
        "commento":"geniale e divertentissimo!",
        "slugute":"fonzie","scorecommento":"4.50"
    },
    {
        "id":"249",
        "datacommento":"2012-07-29",
        "titolocommento":"Bellissima!!!",
        "commento":"..mitighi gli spiriti di mitici brindisi !",
        "slugute":"silviettajp",
        "scorecommento":"4.00"
    },
    {
        "id":"205",
        "datacommento":"2012-06-08",
        "titolocommento":"endecasillabi",
        "commento":"Se fosse pure in endecasillabi sarebbe un cinque!!!
            Complimenti!!",
        "slugute":"Agata",
        "scorecommento":"2.50"
    },
    {
        "id":"198",
        "datacommento":"2012-06-07",
        "titolocommento":"",
        "commento":"meglio difficilmente si pu\u00f2 fare con una sola
           vocale, un tema da seguire, uno humor da rendere.",
        "slugute":"Cosimo de Febrari",
        "scorecommento":"4.00"
    },
    {
        "id":"201",
        "datacommento":"2012-06-07",
        "titolocommento":"",
```

```
52          "commento":"Fichi, grissini, mitili e mirtilli\u2026 una
                composizione in\/di i che fa quasi venire l\u2019acquolina\u
                2026 chapeau!",
53          "slugute":"Robin Claude",
54          "scorecommento":"5.00"
55      },
56      {
57          "id":"196",
58          "datacommento":"2012-06-05",
59          "titolocommento":"Tema adeguato al periodo dell'anno",
60          "commento":"Fa quasi sembrare inutili 4 vocali su 5, trattando
                in modo irridente il tema delle diete.",
61          "slugute":"sandro gasera",
62          "scorecommento":"4.00"
63      }
64  ]
```

Listing 6.4: LoadRecensioni

```
class LoadRecensioni extends AsyncTask<Void,Void,Void> {
2       private JSONParser jParser = new JSONParser();
        @Override
4       protected void onPreExecute() {
            super.onPreExecute();
6           lockScreenOrientation();
        }
8       @Override
        protected Void doInBackground(Void... arg0) {
10          List<NameValuePair> params = new ArrayList<NameValuePair>();
                    params
12              .add(new BasicNameValuePair("id_book",""+id_Book));
            JSONArray json = jParser.makeHttpRequest(
14              ResourceString.HTTPREQUEST_RECENSIONI, "GET", params);
            JSONObject c;
16          for(int i = 0;i<json.length();i++) {
                try {
18                  c = json.getJSONObject(i);
                    myList.add(
20                      new Recensione(
                            c.getInt("id"),
22                          c.getString("slugute"),
                            c.getString("titolocommento"),
24                          c.getString("commento"),
                            (float) (c.getString("scorecommento")
26                          .equals("")?
                            0.0:Float.parseFloat(
28                              c.getString("scorecommento"))),
                            c.getString("datacommento")));
30                  }
                catch (JSONException e) {
32                  e.printStackTrace();
                }
```

```
34                  }
            return null;
36      }
        @Override
38      protected void onPostExecute(Void string) {
            myCustomArrayAdapter =
40              new CustomArrayRecensione(mContext,myList);
            ((TextView)headerView
42              .findViewById(R.id.view_recensioni_count))
                .setText(""+myList.size());
44          myListView.addHeaderView(headerView);
            myListView.setAdapter(myCustomArrayAdapter);
46          if (Build.VERSION.SDK_INT >=
                Build.VERSION_CODES.HONEYCOMB_MR2) {
48                  int shortAnimTime = getResources().getInteger(
                            android.R.integer.config_shortAnimTime);
50                  mRecensione_Loading.animate()
                        .setDuration(shortAnimTime)
52                  .alpha(0)
                    .setListener(new AnimatorListenerAdapter() {
54                      @Override
                        public void onAnimationEnd(Animator animation) {
56                          mRecensione_Loading
                                .setVisibility(View.GONE);
58                      }
                    });
60                  mRecensione_DataLoaded.animate()
                        .setDuration(shortAnimTime)
62                      .alpha(1)
                        .setListener(new AnimatorListenerAdapter() {
64                          @Override
                            public void onAnimationEnd(Animator animation) {
66                              mRecensione_DataLoaded
                                .setVisibility(View.VISIBLE);
68                          }
                        });
70                      }
                    else {
72                      mRecensione_DataLoaded.setVisibility(View.VISIBLE);
                        mRecensione_Loading.setVisibility(View.GONE);
74                  }
            unlockScreenOrientation();
76      }
    }
```

Listing 6.5: JSONParser.java

```
package com.talemotionframework;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONArray;
import org.json.JSONException;

import android.util.Log;

public class JSONParser {

    static InputStream is = null;
    static JSONArray jObj = null;
    static String json = "";

    public JSONArray makeHttpRequest(String url, String method,
            List<NameValuePair> params) {

        // Making HTTP request
        try {

            // check for request method
            if(method == "POST"){
                DefaultHttpClient httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(url);
                httpPost.setEntity(new UrlEncodedFormEntity(params));
                HttpResponse httpResponse = httpClient.execute(httpPost);
                HttpEntity httpEntity = httpResponse.getEntity();
                is = httpEntity.getContent();
            }
            else if(method == "GET"){
                DefaultHttpClient httpClient = new DefaultHttpClient();
                String paramString = URLEncodedUtils
                        .format(params, "utf-8");
```

```
49                  url += "?" + paramString;
                    HttpGet httpGet = new HttpGet(url);
51                  HttpResponse httpResponse = httpClient.execute(httpGet);
                    HttpEntity httpEntity = httpResponse.getEntity();
53                  is = httpEntity.getContent();
                }
55          } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
57          } catch (ClientProtocolException e) {
                e.printStackTrace();
59          } catch (IOException e) {
                e.printStackTrace();
61          }

63          try {
                BufferedReader reader = new BufferedReader(
65          new InputStreamReader(
                                    is, "iso-8859-1"), 8);
67              StringBuilder sb = new StringBuilder();
                String line = null;
69              while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n");
71              }
                is.close();
73              json = sb.toString();
            } catch (Exception e) {
75      //
            }
77
            // try parse the string to a JSON object
79          try {
                jObj = new JSONArray(json);
81          } catch (JSONException e) {
      //
83          }
            return jObj;
85      }
    }
```

Ringraziamenti

Eccoci dunque ai ringraziamenti. Ringrazio per primo me stesso. Me stesso, sì, perchè quest'anno è stato particolare, sono riuscito a riprendermi una cosa che mi stava sfuggendo dalle mani, la mia vita, ho lottato, davvero, ho lottato come meglio potevo, ma ne sono uscito vittorioso. Ringrazio me stesso per essere riuscito ad arrivare fino a qui, per raggiungere questo traguardo e avercela fatta.
Ringrazio la mia famiglia, mia madre, mio padre e mia sorella, li ringrazio perchè mi hanno sempre aiutato, in tutto, mi hanno sostenuto e mi hanno dato la possibilità di arrivare dove sono arrivato. Ringrazio la mia famiglia per avermi accettato per quel che sono e per avermi cresciuto facendomi diventare la persona di cui spero vadano fieri. Vi voglio bene e non avrei potuto sperare di nascere in una famiglia più bella. Vi voglio bene, grazie di tutto.
Ringrazio Alessio, l'amico che c'è sempre stato, che conosco ormai da una vita, con il quale non ci sono stati bassi, ma solo alti, che mi conosce da sempre e che è sempre stato l'amico più caro che ho.
Ringrazio Elettra e Federica, le mie più care amiche, che hanno sopportato questa persona particolare che sono, che mi hanno regalato grandi momenti di felicità e spensieratezza, oltre ad avermi fatto perdere diottrie durante la famosa estate: vi voglio bene.
Ringrazio i miei amici di Università, dai più vecchi ai più recenti: Marco, Marta, Emanuele, Stefano e Andrea con il quale ho condiviso questo percorso lastricato di matrici, integrali e robaccia varia.
Ringrazio i miei amici di collegio Tix, Enrico, Tommy, Ivan, Giulio, Nicola e Loss: compagni di belle serate, di Lan Party e imprechi varii. Ringrazio sopratutto Ivano, per avermi aiutato più e più volte con preparazioni esami, ma ringrazio sopratutto Tommaso e Cif, i quali mi hanno aiutato non tanto ad accettarmi per quello che sono, quanto piuttosto a sbattermene del pensiero altrui (Sì mi piaccioni i pistacchi per dovere di cronaca).
Ringrazio mio cugino Massimo, con il quale sono riuscito ad essere sempre me stesso e con il quale ho passato (e youtube conferma) momenti di stupido divertimento. (Oltre all'ovvio ringraziamento per la correzione tesi).
Ringrazio la mia coinquilina Elena, che adoro ed amo alla follia, con la quale siamo riusciti, dopo molti tentativi, ad invocare finalmente il dio del disagio. Ti voglio bene tesoro.
Ringrazio Riccardo, per avermi rotto tutto quest'anno ma per essere la persona, in qualche modo gradevole, che è. Ringrazio Gioia che si nutre al mio stesso modo.
Ringrazio Silvano e Sandro, istruttori e amici di palestra, che quest'anno hanno fatto un gran lavoro sulla mia persona e mi hanno aiutato e ritornare sulla giusta via.
Ringrazio Sergio, Pippo, Francy e Scott per le splendide persone che sono sempre state con i quali ho passato davvero momenti importanti.
Per ultimi ho lasciato voi due, Michele e Glauco. Vi ringrazio: quest'anno ci siete sempre stati, con tutti i problemi che ho avuto, mi avete aiutato a superarli, bastonandomi se era necessario, ma sempre presenti, a sopportarmi e supportarmi, vi voglio bene e sono contento e fiero di avervi conosciuto. Siete molto importanti e vi ringrazio per tutto.
Siamo giunti alla fine. Vi ringrazio nuovamente tutti. Grazie per esserci stati.

Antonio Davide Calì