

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

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

**Progettazione e sviluppo di un'applicazione  
smartphone per il turismo alberghiero**

Relazione finale in  
Mobile Web Design

Relatore  
Dott. Mirko Ravaioli

Presentata da  
Eros Gobbi

Sessione III  
Anno Accademico 2012 - 2013



# Indice

<b>Introduzione</b> .....	<b>7</b>
<b>1 - I dispositivi mobile</b> .....	<b>9</b>
<b>1.1 - Storia dei dispositivi mobili</b> .....	<b>11</b>
<b>1.2 - Sistemi operativi per dispositivi mobili</b> .....	<b>16</b>
1.2.1 - <i>iOS</i> .....	17
1.2.2 - <i>Android</i> .....	18
1.2.3 - <i>Windows Phone</i> .....	20
1.2.4 - <i>BlackBerry</i> .....	22
1.2.5 - <i>Symbian</i> .....	23
<b>2 - Progetto</b> .....	<b>25</b>
<b>2.1 – Progettazione applicazione</b> .....	<b>26</b>
2.1.1 - <i>Splash Screen</i> .....	26
2.1.2 - <i>Home</i> .....	27
2.1.3 - <i>Griglia e Gallery</i> .....	28
2.1.4 - <i>Mappa</i> .....	29
2.1.5 - <i>Offerte</i> .....	31
2.1.6 - <i>Dettaglio Offerta</i> .....	32
2.1.7 - <i>Prenota</i> .....	33
2.1.8 - <i>Last Minute</i> .....	34
<b>2.2 - Progettazione server</b> .....	<b>35</b>
2.2.1 - <i>Struttura JSON</i> .....	35
<b>2.3 - Progettazione sito web per amministratore</b> .....	<b>36</b>
2.3.1 – <i>Crea nuova offerta</i> .....	36
2.3.2 – <i>Visualizza Prenotazioni</i> .....	36
<b>3 - Implementazione della soluzione su Android</b> .....	<b>37</b>
<b>3.1 – Implementazione applicazione</b> .....	<b>37</b>
3.1.1 - <i>Classe SplashScreenActivity</i> .....	38
3.1.2 - <i>Classe Home</i> .....	39
3.1.3 - <i>Classe GestioneFooter</i> .....	40
3.1.4 - <i>Classe Griglia</i> .....	41
3.1.5 - <i>Classe Gallery</i> .....	42
3.1.6 - <i>Classe Maps</i> .....	43
3.1.7 - <i>Classe MyOverlay</i> .....	45
3.1.8 - <i>Classe JsonParser</i> .....	47

3.2 - Implementazione database .....	48
3.3 - Implementazione sito web per amministratore .....	50
4 – Sviluppi futuri .....	55
5 – Conclusioni.....	57
6 – Bibliografia .....	59

# *Indice delle figure*

<i>FIGURA 1 - EVOLUZIONE DEI DISPOSITIVI MOBILE</i> .....	11
<i>FIGURA 2 - EVOLUZIONE DEI TELEFONI CELLULARI</i> .....	15
<i>FIGURA 3 - SPLASH SCREEN</i> .....	26
<i>FIGURA 4 - HOME</i> .....	27
<i>FIGURA 5 - GRIGLIA FOTO</i> .....	29
<i>FIGURA 6 – FOTO A SCHERMO INTERO</i> .....	29
<i>FIGURA 7 – CONDIVISIONE FOTO</i> .....	28
<i>FIGURA 8 - ATTIVAZIONE GPS</i> .....	29
<i>FIGURA 9 - IMPOSTAZIONI DI LOCALIZZAZIONE</i> .....	29
<i>FIGURA 10 - TERMINI DI UTILIZZO</i> .....	29
<i>FIGURA 11 - ITINERARIO PER L'HOTEL</i> .....	30
<i>FIGURA 12 - MESSAGGIO ALLA PARTENZA</i> .....	30
<i>FIGURA 13 - MESSAGGIO ALL'ARRIVO</i> .....	30
<i>FIGURA 14 - LISTA DELLE OFFERTE</i> .....	31
<i>FIGURA 15 - DETTAGLIO OFFERTA</i> .....	32
<i>FIGURA 16 – FORM PER LA PRENOTAZIONE</i> .....	33
<i>FIGURA 17 - LAST MINUTE</i> .....	34
<i>FIGURA 18 - CREA NUOVA OFFERTA</i> .....	36
<i>FIGURA 19 - VISUALIZZA PRENOTAZIONI</i> .....	36
<i>FIGURA 20 - FILE DI PROGETTO</i> .....	37
<i>FIGURA 21 - CLASSE SPLASHSCREENACTIVITY</i> .....	38
<i>FIGURA 22 - CLASSE BACKGROUNDASYNCCTASK</i> .....	38
<i>FIGURA 23 - METODO ONPREEXECUTE</i> .....	38
<i>FIGURA 24 - METODO DOINBACKGROUND</i> .....	39
<i>FIGURA 25 - METODO ONPOSTEXECUTE</i> .....	39
<i>FIGURA 26 - CLASSE HOME</i> .....	39
<i>FIGURA 27 - CLASSE GESTIONEFooter</i> .....	40
<i>FIGURA 28 - CLASSE GRIGLIA</i> .....	41

<i>FIGURA 29 - CLASSE GALLERY</i> .....	42
<i>FIGURA 30 - CLASSE MAPS</i> .....	43
<i>FIGURA 31 - METODO ONCREATE</i> .....	43
<i>FIGURA 32 - METODO ONCREATE PARTE 2</i> .....	44
<i>FIGURA 33 - CLASSE MYOVERLAY</i> .....	45
<i>FIGURA 34 - METODO DRAW</i> .....	46
<i>FIGURA 35 - CLASSE JSONPARSER</i> .....	47

# Introduzione

La progettazione e lo sviluppo di una applicazione Android, con la quale gli utenti possono visionare e prenotare le offerte di un hotel sarà l'argomento di discussione su cui verterà questa tesi.

Lo sviluppo tecnologico permette al giorno d'oggi, a qualunque attività commerciale, di potersi pubblicizzare attraverso svariate piattaforme. Ciò comprende il mercato degli smartphones che è ritenuto quello in maggiore crescita negli ultimi anni. Questa necessità comprende ovviamente anche il settore alberghiero e notando una carenza di applicazioni Android che offrono questo servizio l'idea è stata quella di offrire, proprio per dispositivi che utilizzano questo sistema operativo, una serie di funzionalità volte alla presentazione di un hotel.

Per la realizzazione di questo progetto sono stati sviluppati un applicativo per dispositivi mobile, alcune pagine web e il web service che si interfaccia col client.

Nel primo capitolo vengono introdotti i concetti di base volti a garantire una maggiore comprensione del progetto che sarà presentato. Viene proposta una breve panoramica sugli smartphones, con accenni alla storia di quest'ultimi. Sono successivamente descritti i principali sistemi operativi oggi disponibili sul mercato.

Nel secondo capitolo viene presentata l'applicazione, ne vengono descritte nel dettaglio le funzionalità e le modalità di utilizzo.

Nel terzo capitolo sono descritti i dettagli implementativi dell'applicazione, del database e del sito web. Vengono presentate le classi costituenti il progetto, con particolare attenzione ai principali metodi che queste implementano.

Nel quarto capitolo sono considerati i futuri sviluppi dell'applicazione, con una breve trattazione sui possibili miglioramenti e sulle nuove funzionalità apportabili al progetto.

Nel quinto ed ultimo capitolo vengono affrontate le conclusioni sottolineando le conoscenze didattiche apprese sviluppando il progetto.





# 1 - I dispositivi mobile

È impossibile prevedere ciò che accadrà in futuro ma si può fare di esso una proiezione che può assomigliarci molto. Molti studiosi nel campo informatico affermano che tra qualche anno non esisterà più il concetto di Computer Desktop<sup>[1]</sup> ma che ci sarà la possibilità di avere integrato in oggetti di uso quotidiano l'elaborazione delle informazioni e l'accesso alla rete, a volte anche senza essere coscienti che ciò sta accadendo. Mark Weiser, creatore di questo modello chiamato Ubiquitous Computing<sup>[2]</sup>, affermò che questa sarà la terza ondata per quanto riguarda i computer: inizialmente c'erano i mainframe condivisi da molte persone, ora ci sono i personal computer, persone e macchine che si fissano l'un l'altro, poi verrà l'Ubiquitous Computing cioè quando la tecnologia farà solo da sfondo alle nostre vite.<sup>[3]</sup>

Una svolta in questo senso la stiamo già vivendo con i numerosi dispositivi mobili che sono in circolazione. La presenza di 1,8 miliardi di connessioni 3G e di quasi sei miliardi di dispositivi connessi alla rete mobile la dice lunga sul quale possa essere molto probabilmente il salto che si farà in avanti<sup>[4]</sup>. I dispositivi mobili hanno cambiato le nostre vite e la loro evoluzione è tale che oggi un loro utilizzo senza funzionalità quali la telefonia vocale, il text messaging e il mobile Internet sarebbe praticamente impensabile.

È doveroso però suddividerli in varie categorie anche se spesso è difficile tracciare confini netti.<sup>[5]</sup>

**PDA.** I Personal Digital Assistant, meglio conosciuti come palmari, sono strumenti che possono essere tenuti in una mano e possono risiedere comodamente nella tasca del proprietario. Sono principalmente diffusi in ambito aziendale, perché utili per annotare promemoria, numeri di telefono e appuntamenti ma sono stati impiegati con successo anche in ambito turistico. Recentemente, pur essendo in calo le vendite dal momento che le stesse funzionalità sono offerte dai cellulari, hanno ricevuto uno slancio commerciale grazie all'integrazione di un modulo GPS e di connettività wireless.

**Tablet PC.** Sono strumenti delle dimensioni di un block notes cartaceo, tendenzialmente privi di tastiera e possiedono uno schermo sensibile al tocco. Il loro utilizzo sta comprendendo sempre più l'ambito personale (gioco, visione di fotografie e

di filmati), quando inizialmente furono concepiti per un uso aziendale. Sono dispositivi a metà strada tra notebook e smartphone anche se più orientati verso quest'ultimi.

L'attenzione sarà però rivolta verso gli **Smartphones**. Si tratta di dispositivi mobili che abbinano le caratteristiche di un telefono cellulare a quelle di gestione dati personali<sup>[6]</sup>. La principale caratteristica però è la possibilità di installare ulteriori applicazioni, dette App, che aggiungono nuove funzionalità. Queste possono essere sviluppate dal produttore dello smartphone, dall'utilizzatore o da terze parti e vengono rilasciate sul market online gratuitamente o a pagamento. Nel mondo gli smartphone rappresentano oltre il 40% del mercato.

Tutti questi dispositivi condividono diverse caratteristiche: modulo GPS per la localizzazione del dispositivo, connettività alla rete, e-mail, fotocamera, alimentazione a batteria. Ma gli smartphone sono maggiormente performanti degli altri per diversi motivi. I processori sono più evoluti e le memorie sempre più capienti, molti hanno il touchscreen che facilita ulteriormente l'interfaccia e hanno dei sistemi operativi sviluppati ad hoc quali Android, Apple iOS, Windows Phone, Blackberry OS, Symbian OS, Ubuntu.

# 1.1 - Storia dei dispositivi mobili

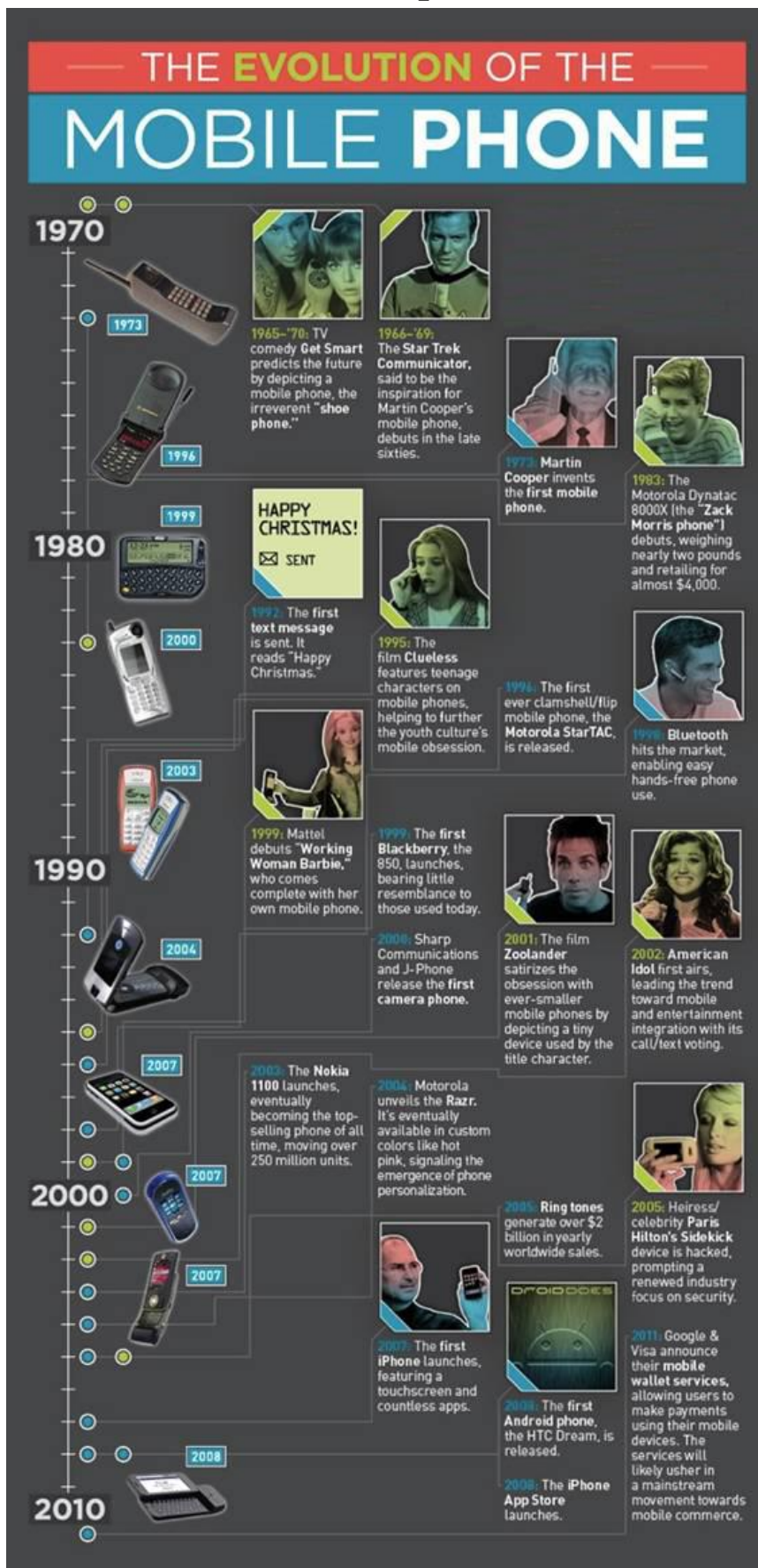


Figura 1 - Evoluzione dei dispositivi mobile

Inizialmente fu AT&T a commercializzare, nel 1947, il primo servizio che permise di far comunicare due dispositivi mobili. L'MTS (Mobile Telephone Service) arrivò ad essere presente in più di cento città e ad essere utilizzato da oltre 5000 clienti. Le chiamate erano gestite da un operatore e l'utente doveva premere un tasto per parlare e rilasciarlo per ascoltare. Essendoci solo tre canali radio le telefonate in contemporanea potevano essere appunto al massimo tre. Il kit per la chiamata pesava 36 Kg e il servizio costava 15\$ al mese più 4.75\$ per ogni chiamata effettuata. Una soluzione alternativa, creata per competere con AT&T fu Radio Common Carriers (RCC) disponibile anche su telefoni full-duplex trasportabili in apposite valigette. Il primo vero telefono mobile, noto come Mobile Telephone system A (MTA), fu introdotto in Svezia nel 1956. Permetteva di ricevere ed effettuare chiamate da un'auto, senza però mai cambiare la cella di riferimento durante la chiamata altrimenti si sarebbe persa la connessione. Per ragioni ancora sconosciute il sistema fu spento alla fine del 1959 e mai più riattivato.

Bisogna arrivare al 1973<sup>[7]</sup> per vedere implementato il primo telefono mobile. Martin Cooper, un ricercatore Motorola, effettuò la prima telefonata da dispositivo mobile chiamando Joel Engel dei Laboratori Bell. L'apparecchio pesava 1.1 Kg e necessitava 10 ore di carica per 30 minuti di chiamata.

**1G.** È quella che ha visto nascere una infrastruttura di trasmettitori che ospitavano le celle telefoniche e che erano tra loro collegati in modo da permettere agli utenti di usare il loro dispositivo anche in viaggio. Il Giappone nel 1980 costruì il proprio network che fu il primo a coprire per intero un territorio nazionale. Seguì poi la società telefonica danese Nordic Mobile Telephone (NMT) che operando in Danimarca, Svezia, Finlandia e Norvegia fu la prima ad offrire il roaming internazionale.

**2G.** Vede la luce nel 1990 con l'arrivo di due nuove tecnologie digitali, l'europea GSM e l'americana CDMA. Fu possibile introdurre la messaggistica SMS (primo SMS nel 1992) e servizi di download (ad esempio le suonerie). Il primo Smartphone in assoluto, chiamato Simon, fu progettato dalla IBM nel 1992 e commercializzato dalla BellSouth a partire dal 1993. Oltre alle comuni funzioni di telefono incorporava calendario, rubrica, orologio, block notes, e-mail e giochi: per poter scrivere direttamente sullo schermo era disponibile un pennino. Non ebbe molto successo a causa del costo proibitivo e della clientela ancora poco propensa a comprendere l'utilità di un telefono "tuttofare".

Nel 1997 viene lanciato sul mercato PalmPilot che dà il via alla seconda generazione di palmari Palm. Esclusa la funzione di telefono di cui PalmPilot è sprovvisto, presenta comunque una serie di funzionalità quali rubrica, blocco note, calendario e agenda che ne incentivano la rapida diffusione soprattutto in ambienti lavorativi. Nello stesso anno

siccome non era più sufficiente una sola linea per gestire tutto il traffico, fu creata una seconda banda, la 1800MHz dando vita dunque ai telefonini Dual Band. Quello successivo, è l'anno del Nokia 9110 che a tutti gli effetti si avvicina, per funzionalità, agli smartphones odierni e catapulta Nokia tra i leader del settore di telefonia. È alla fine degli anni '90 che arriva però la vera rivoluzione. L'azienda canadese RIM, nota per la produzione di "cercapersone" lancia sul mercato un telefono in grado di navigare in rete e leggere mail. Nasce il BlackBerry 5810. Ciò da inizio a tutti gli effetti all'era degli smartphones.

**3G.** Nasce ad ottobre 2001 in Giappone con il servizio di Mobile Internet NTT DoCoMo e si diffuse prima in US e Sud Corea e poi in Europa (2003). La terza generazione trasformò radicalmente il mondo della telefonia mobile rendendo possibile un uso diffuso dei dispositivi mobili per l'accesso al web e per la trasmissione di servizi televisivi e radiofonici. Nel 2003 gli schermi diventano a colori e con un livello di definizione sempre più elevato.

E' con il 3G che si affermano gli smartphones.

In questo periodo si è passati da dispositivi grandi, scomodi da usare, pesanti, sempre bisognosi di carica, poco pratici da utilizzare a device sempre più sottili, trasportabili, con display grandi e ad alta risoluzione.

La prima ad intuire questo cambiamento è, nel 2007, Apple che presenta il primo iPhone. L'azienda di Cupertino entra ufficialmente, con questo prodotto, nel mercato della telefonia, concependo un prodotto dagli standard qualitativi molto elevati, ricco di funzionalità e semplice da usare. Il dispositivo utilizza iPhone OS, più tardi conosciuto con il nome di iOS, una versione ottimizzata di Mac OS X. Include fotocamera digitale<sup>[8]</sup>, lettore multimediale e, oltre ai normali servizi di telefonia, permette di utilizzare servizi come navigazione web, e-mail e di accedere alla rete tramite Wi-Fi. È dotato di uno schermo multi-touch molto sensibile e preciso, in grado di fornire all'utente una nuova esperienza di utilizzo, di accelerometro e giroscopio, che fungono da sensori di movimento, un sensore di prossimità e un sensore di luce ambientale.

Alla fine dello stesso anno Google sviluppa un nuovo sistema operativo: Android, che è destinato a diventare il sistema operativo di riferimento per i sostenitori dell'open source. Nell'ottobre 2008 viene lanciato sul mercato il primo smartphone basato su sistema operativo Android. Il dispositivo in questione è l'HTC Dream. Seguiranno poi nuovi prodotti sempre più sofisticati anche da parte di Samsung e della stessa HTC che consolideranno la collaborazione con Google.

Nel 2010 anche Windows mette a punto un nuovo sistema operativo, Windows Phone 7, che in Italia si conoscerà per la prima volta grazie a LG, con il suo LG Optimus 7. Il 3 settembre 2013 viene annunciato ufficialmente un accordo dal valore di 5,44 miliardi di euro con il quale Microsoft si aggiudica la divisione Devices & Services di Nokia. Ciò fa presagire ad un'integrazione di Windows Phone anche per i futuri prodotti della casa finlandese.

**4G.** Non ancora definita come uno standard diffuso, nasce dall'arrivo sul mercato da tecnologie come il WiMAX di Sprint e come l'LTE. Il 4G segna il passaggio a reti IP native e trasforma internet mobile in qualcosa di molto simile all'internet usato da casa su computer fissi. La quarta generazione mobile è ancora in fase evolutiva, sicuramente il più grande vantaggio della nuova tecnologia è la velocità, dieci volte superiore a quella della tecnologia 3G.



Figura 2 - Evoluzione dei telefoni cellulari

## **1.2 - Sistemi operativi per dispositivi mobili**

Nel paragrafo precedente si è accennato a sistemi operativi come iOS e Android, per citarne un paio, ma cos'è un sistema operativo? E quali sono le differenze tra le principali piattaforme oggi in commercio? Usando una definizione piuttosto semplice e senza entrare nei dettagli, in informatica con sistema operativo<sup>[9]</sup> si intende l'insieme del software di base installato su un dispositivo indispensabile al funzionamento dello stesso. Esso ha quindi lo scopo di gestire risorse hardware e software dell'elaboratore, fare da interfaccia tra utente e hardware e consente di estendere le funzionalità del dispositivo stesso installando nuovi software appositamente progettati per essere riconosciuti dal sistema operativo stesso. Tuttavia affronta problematiche legate alla natura del dispositivo mobile, più critiche rispetto ad un desktop o un laptop, per esempio: la limitatezza delle risorse (memoria, CPU), l'assenza di alimentazione esterna, differenti protocolli di trasferimento dati per l'accesso a internet (WiFi, GPRS, HSDPA...), nuovi metodi d'immissione (touchscreen, minitastiere) e le ridotte dimensioni del display.

Oggi i sistemi operativi più diffusi presenti sul mercato sono cinque: Apple iOS, Android, Windows Mobile, Blackberry OS, Symbian OS; nel capitolo seguente illustriamo le caratteristiche di ciascun SO.



### 1.2.1 - iOS

Questo sistema operativo ha il merito di aver dato il via alla rivoluzione del settore mobile. Rilasciato per la prima volta il 29 giugno 2007<sup>[10]</sup> e conosciuto fino al 2010 con il nome di “iPhone OS” è stato sviluppato da Apple per iPhone, iPad e iPod touch e ne è stata rilasciata nel 2013 la versione 7. In pieno stile Apple è un sistema poco personalizzabile, utilizzabile esclusivamente dai propri dispositivi, a differenza di Windows Phone e Android che supportano hardware prodotto da terzi. iOS è stato il capostipite assoluto della nuova generazione di sistemi operativi mobile touch-based e app-based, introducendo tramite l’App Store il concetto di “negoziato online” da cui è possibile scaricare ed installare nuove applicazioni, gratuite o a pagamento, per iPhone, iPod touch e iPad. Le App possono essere realizzate da terzi, attraverso l’utilizzo dell’apposito SDK rilasciato il 6 marzo 2008. Esso permette agli sviluppatori di creare applicazioni e testarle in un simulatore; tuttavia il caricamento di una applicazione nei dispositivi è possibile solamente dopo aver pagato una tassa di iscrizione all’iPhone Developer Program. L’ambiente di sviluppo per l’iPhone SDK è Xcode, che è lo stesso che consente lo sviluppo di applicazioni per Mac OS X. I linguaggi supportati da Xcode sono L’Objective C, l’AppleScript, il C++, l’Objective C++ e Java. Gli sviluppatori sono liberi di scegliere qualsiasi prezzo per le loro applicazioni che sono distribuite tramite App Store, per le quali riceveranno il 70% del ricavo. Essi possono anche optare per rilasciare l’applicazione gratis e non pagheranno nessun costo di rilascio o distribuzione, eccetto la tassa di sottoscrizione al programma developer. Prima di essere propriamente pubblicate le App dovranno essere state validate da Apple stessa. Questi rigidi controlli da parte di Apple da un lato consentono di garantire la costante presenza sull’App Store di App dagli standard qualitativi piuttosto elevati. Dall’altro lato Apple ha la possibilità di controllare e di negare, eventualmente, la pubblicazione a quelle applicazioni che entrerebbero eccessivamente in concorrenza con i servizi offerti dall’azienda di Cupertino. Con iOS viene introdotto anche il sistema di notifiche push. Quest’ultime costituiscono una funzione chiave dell’iPhone e permettono di ricevere avvisi in tempo reale da parte di un’app, sotto forma di messaggi popup, anche quando l’applicazione stessa non è in esecuzione. Con l’avvento di iOS4 sono state introdotte le funzioni Multitasking che consentono di tenere aperte più applicazioni contemporaneamente, di passare dall’una all’altra ed eseguire App in background.

### 1.2.2 - Android

Android è un sistema operativo open source<sup>[11]</sup> per dispositivi mobili basato su kernel 2.6 di Linux. Nasce nel 2003 per mano di Android Inc. che viene poi acquisita da Google nel 2005. Il primo smartphone dotato di piattaforma Android è stato l'HTC Dream presentato il 22 ottobre del 2008. Android è la prima piattaforma completa e gratuita per lo sviluppo di applicazioni mobile. Ha il vantaggio di essere supportato da una vasta gamma di dispositivi, non necessariamente smartphones ma anche tablet, e-book reader e netbook, a differenza di iOS che, come descritto precedentemente, è installabile esclusivamente su dispositivi Apple. Questa importante caratteristica ne ha favorito in breve tempo una rapida distribuzione e l'ha reso, oggi giorno, il sistema operativo più diffuso. I nomi delle varie versioni del sistema operativo richiamano nomi di dolci e le iniziali sono in ordine alfabetico. Abbiamo quindi: Cupcake (versione 1.5), Donut (versione 1.6), Eclair (versione 2.0 e 2.1), Froyo (versione 2.2), Gingerbread (versione 2.3), Honeycomb (versioni 3.0, 3.1 e 3.2), Ice Cream Sandwich (versione 4.0) e l'ultima in commercio Jelly Bean (versioni 4.1, 4.2 e 4.3). È stata annunciata anche la prossima che si chiamerà Kit Kat (versione 4.4).

Anche Android ha un proprio store di applicazioni online, conosciuto con il nome di Android Market, diventato successivamente Play Store, che conta un ampio numero di applicazioni. I dispositivi Android possono eseguire applicazioni scritte da sviluppatori di terze parti e distribuite attraverso il Play Store. Una volta firmate<sup>[12]</sup>, è data la possibilità allo sviluppatore di rendere le applicazioni realizzate immediatamente disponibili, senza dover passare per lunghi processi di attivazione. Questo può essere un vantaggio in termini di tempi ma la quasi totale assenza di rigidi controlli aumenta notevolmente la possibilità di imbattersi nel download di applicazioni poco affidabili. Android ha però il vantaggio di poter essere altamente personalizzabile. L'utente può combinare tra loro app, widget e scaricare nuove interfacce direttamente dal Play Store.

Le applicazioni, una volta scritte in linguaggio Java, vengono eseguite tramite la Dalvik Virtual Machine, una macchina virtuale adattata per l'uso su dispositivi mobili dotata di compilatore just-in-time (JIT) il quale traduce il bytecode nel codice macchina nativo in fase di run-time. La DVM grazie ad un utilizzo intelligente dei registri di sistema permette una maggiore ottimizzazione della memoria in dispositivi con bassa capacità, consente di far girare diverse istanze della macchina virtuale contemporaneamente e nasconde al sistema operativo sottostante la gestione della memoria e dei thread. Le applicazioni per Android si sviluppano in Java per poi trasformare il bytecode Java in bytecode Dalvik, quindi su un dispositivo Android non girerà alcun bytecode Java ma un bytecode le cui specifiche sono descritte dal formato .dex (Dalvik EXecutable).

Android é fornito di una serie di applicazioni preinstallate: un browser, una rubrica e un calendario. Quando si vuole aggiungere una funzionalità non presente si usa ricercarla nello Store e "installarla", ossia copiarla all'interno del dispositivo. Dalla versione 2.2 di Android é possibile installare una app, oltre che nella memoria interna del dispositivo, su una card esterna. Il software viene distribuito sotto forma di pacchetto autoinstallante, quindi un file con estensione .APK. Questo non é altro che un file compresso, contenente il software (file con estensione .dex) le sue risorse (immagini, suoni ecc...) e alcuni file XML. La struttura del framework é molto chiara se si utilizza l'ambiente di sviluppo SDK (Software Development Kit), esso include gli strumenti di sviluppo, le librerie, un emulatore del dispositivo, la documentazione, alcuni progetti di esempio, tutorial e altro. È installabile su qualsiasi computer che usi come sistema operativo Windows, Mac OS X o Linux. L'IDE (Integrated Development Environment) ufficialmente supportato per lo sviluppo di applicazioni per Android é Eclipse, per cui é fornito un plug-in progettato per fornire un ambiente potente ed integrato in cui costruire le applicazioni. Le applicazioni Android sono caratterizzate da una certa dualità: parti dinamiche scritte in Java e parti statiche scritte in XML. Tipico delle parti statiche possono essere quelle caratteristiche che non cambiano durante l'esecuzione dell'applicazione, come per esempio il colore dello sfondo; tipico delle parti dinamiche sono invece gli aspetti programmatici come per esempio la gestione degli eventi.

Ai fini della programmazione, il team di Android ha specificato nella documentazione ufficiale vari termini per definire i vari tipi di applicazioni:

**Activity.** sono quelle applicazioni destinate a una interazione diretta con l'utente utilizzando lo schermo e i dispositivi di input messi a disposizione dallo smartphone.

**Service.** sono, al contrario, quelle applicazioni che per loro natura svolgono delle operazioni autonome e che vengono richiamati dalle attività al bisogno; gira in sottofondo e non interagisce direttamente con l'utente. Un esempio è la tastiera che compare quando si tappa sul campo di input testuale.

**Fragment.** È quella porzione di codice (quindi di applicazione) che gestisce la parte grafica, in base alle possibilità del dispositivo su cui è stato installato.

### 1.2.3 - Windows Phone

Il lancio di Windows Phone 8<sup>[13]</sup> avviene nell'ottobre 2012 e segna un importante passaggio nella piattaforma mobile di Microsoft che viene completamente rinnovata rispetto alle precedenti versioni sia nell'interfaccia utente sia per quanto riguarda gli strumenti di sviluppo. I primi smartphones con sistema operativo Windows Phone presentano, oltre a capacità di elaborazione migliorate, il supporto multitouch e una nuova interfaccia grafica. L'11 febbraio del 2011, durante una conferenza stampa a Londra, l'amministratore delegato della Microsoft Steve Ballmer e l'amministratore delegato della Nokia Stephen Elop hanno annunciato una partnership tra le loro aziende nella quale Windows Phone diventerà il principale sistema operativo della Nokia, Bing sarà il motore di ricerca e sarà presente un'integrazione tra Nokia Maps con Bing Maps. Nel 2013 Microsoft acquisisce la divisione Devices & Services di Nokia. La principale innovazione a livello grafico portata dal sistema operativo di Redmond riguarda le "Live Tiles", dei riquadri simili ad icone, altamente personalizzabili anche nelle dimensioni, che fungono da "raccoltori" di contenuti. Sotto ogni singola "piastrella" è infatti possibile aggregare applicazioni e dati. Oltre ai miglioramenti grafici Windows Phone 8 dal lato hardware introduce il supporto ai processori multicore, supporta gli schermi capacitivi multitouch, il supporto per schede MicroSD e compatibilità con display di diverse dimensioni. Si può dire che WP8 è il perfetto mix tra la vasta selezione di Android di fattori di forma e lo stretto controllo di Apple OS sull'integrazione hardware. Microsoft controlla essenzialmente quasi tutto questa volta, dalla risoluzione minima dello schermo, alla quantità di ram e la velocità dell'interfaccia. Anche per Windows Phone esiste uno store online, il Windows Phone Store (precedentemente chiamato Marketplace). Questo servizio viene usato per distribuire e vendere musica, video e applicazioni per il sistema operativo di casa Microsoft. Gli sviluppatori registrati a Windows Phone 8 ed Xbox Live possono inserire e modificare le loro applicazioni per la piattaforma attraverso l'applicazione online App Hub. L'App Hub fornisce strumenti di sviluppo e di supporto per gli sviluppatori di applicazioni; le applicazioni inserite devono sottostare all'approvazione e la validazione di Microsoft. Il costo delle applicazioni approvate dipende dallo sviluppatore, ma a Microsoft spetta il 30% del guadagno (il restante 70% va agli sviluppatori). Per far parte della lista ufficiale gli sviluppatori devono pagare una tassa annuale. Applicazioni e giochi per WP8 sono basati su XNA<sup>[14]</sup> o su specifiche versioni di Microsoft Silverlight. Microsoft XNA (XNA is Not an Acronym) è un framework realizzato da Microsoft specifico per lo sviluppo di videogiochi e applicazioni su più piattaforme: PC, Xbox 360 e Windows Phone 8. Esso si occupa di fornire una serie di strumenti (sia utility che classi) atti a semplificare notevolmente lo sviluppo di videogiochi sia in 2D che in 3D. XNA ha

l'abilità infatti di coprire molti degli aspetti che normalmente bisogna affrontare durante lo sviluppo di un videogioco, ad esempio offrire un'implementazione integrata per la gestione della grafica, del suono e dell'input. Basato su Direct X 9 e .NET Framework, esso si sviluppa in C# con l'utilizzo dell'IDE Visual Studio. Un'altra caratteristica molto apprezzata di XNA é la sua possibilità di poter lavorare sia ad alto livello che a basso livello, a discrezione dello sviluppatore.

Microsoft offre inoltre il Windows Phone SDK un pacchetto software gratuito per sviluppare applicazioni per Windows Phone 8 che contiene:

- Visual Studio 2010 Express e Expression Blend per Windows Phone
- Windows Phone Emulator
- Silverlight per Windows Phone
- Microsoft Expression Blend per Windows Phone
- XNA Game Studio 4.0

#### **1.2.4 - BlackBerry**

BlackBerry OS<sup>[15]</sup> è un sistema operativo mobile proprietario, sviluppato dalla canadese RIM (Research In Motion) in C++ per la sua linea di smartphones BlackBerry. Fornisce multitasking e supporta dispositivi di input quali la trackball e, più recentemente, trackpad e touchscreen. BB10 è l'ultimo sistema operativo per smartphone e tablet in ordine di uscita presentato da BlackBerry il 30 Gennaio 2013. Ma la principale differenza con gli altri sistemi operativi è forse il BlackBerry MDS (Mobile Data System): è un sistema che permette l'accesso al BES (BlackBerry Enterprise Server) che è uno dei servizi che consente ad un'azienda di sincronizzare i propri server di posta o di applicazioni con i cellulari in dotazione ai dipendenti, i quali possono ricevere nella propria casella di posta le e-mail tramite un canale wireless sicuro. Il sistema offre inoltre una serie di funzioni sulla sicurezza, tra cui la cifratura dei dati e il supporto alle smart card, utile se si vuole, ad esempio, aggiungere una firma digitale alle mail. Il market delle applicazioni per il sistema operativo di RIM si chiama BlackBerry App World. Il servizio offre agli utenti di BlackBerry, un ambiente per navigare, scaricare e aggiornare le applicazioni sviluppate da terze parti; il servizio è online dal 1 aprile 2009. Le applicazioni sono sia gratuite che a pagamento a partire da 0,99 \$ fino a 999,99 \$ USD. Gli sviluppatori che intendono scrivere applicazioni per i dispositivi BlackBerry hanno a disposizione da RIM essenzialmente due scelte: la prima è utilizzare un ambiente di sviluppo specifico creato ad hoc da RIM, il BlackBerry JDE (BlackBerry Java Development Environment). Permette di creare, importare ed esportare progetti BlackBerry, nonché di compilarli ed eseguirli o effettuarne il debug su dispositivi collegati al calcolatore. La seconda scelta possibile è quella di utilizzare Eclipse, un potente IDE per applicazioni in Java, ed in altri molteplici linguaggi di programmazione come C, C++, Python, PHP e altre. Eclipse è plugin extensible e grazie a questa caratteristica RIM mette a disposizione un plugin per godere, anche in questo ambiente, degli stessi servizi presenti nel suo JDE nativo. Per l'effettivo debugging delle applicazioni vengono forniti diversi simulatori che riproducono, anche graficamente, le sembianze e le funzionalità dei dispositivi BlackBerry.

Inoltre è possibile effettuare il porting per alcune delle applicazione (viene stimato circa il 65%) studiati e sviluppati per Android tramite il "BlackBerry Runtime for Android Apps", strumento a uso gratuito.

### 1.2.5 - Symbian

Symbian OS è un sistema operativo per smartphones prodotto da Symbian Foundation<sup>[16]</sup> un'organizzazione non-profit composta da Nokia, Sony Ericsson, Motorola, Texas Instruments, Samsung e altri che ha il compito di gestire la piattaforma. Nel 2013, Nokia ha comunicato di non voler più utilizzare Symbian per i propri dispositivi e di terminare la produzione di dispositivi dotati di questo SO. Lo sviluppo di Symbian OS è attualmente supportato da Accenture, ma, dopo l'abbandono da parte di Nokia, nessun hardware lo supporta più. Symbian<sup>[17]</sup> dispone di funzionalità di multithreading, multitasking e protezione della memoria. Grande importanza è data all'utilizzo della memoria, mediante tecniche specifiche di Symbian che determinano la rarità degli errori dovuti a una cattiva gestione della stessa (memory leak). Tecniche analoghe permettono di garantire un'efficiente gestione dello spazio su disco e un corretto uso della CPU, consentendo così di preservare anche la durata della batteria. Per Symbian OS è disponibile un discreto numero di programmi, sia gratuiti che a pagamento, il che ne fa un prodotto espandibile e personalizzabile. La piattaforma che permette di scaricare nuove applicazioni per i dispositivi Symbian si chiama Ovi Store. L'Ovi Store è stato lanciato in tutto il mondo nel maggio 2009; qui, i clienti possono scaricare giochi per cellulari, applicazioni, video, immagini e toni di chiamata per i propri dispositivi Nokia. Per gli sviluppatori, Nokia offre uno strumento self-service per portare i loro contenuti sull'Ovi Store. I tipi di contenuto supportati comprendono: Java ME, applicazioni Flash, widget, suonerie, sfondi, temi e molto altro ancora. Nonostante la recente decisione di Nokia di utilizzare Windows Phone 8 come sistema operativo primario, Ovi Store sarà ancora disponibile per il presente e il futuro dei telefoni Symbian, mentre Ovi Store e Windows Phone Store verranno uniti sulla piattaforma WP 8. Esistono svariati linguaggi per l'implementazione di software per Symbian, ma i principali sono due:

**Symbian C++.** Il linguaggio nativo del sistema operativo Symbian, pur non rappresentando lo standard d'implementazione. Per realizzare un'applicazione per Symbian in tale linguaggio sono necessari alcuni prerequisiti, da installare nel proprio PC:

- SDK (Software Development Kit), utile strumento per lo sviluppo e il test dell'applicazione in C++.
- IDE (Integrated Development Environment) è letteralmente un ambiente di sviluppo integrato, costituito da una famiglia di software, destinato a facilitare la creazione dell'applicazione. Un esempio ne è Carbide.

- C++, sistema basato su Eclipse che permette di utilizzare un'interfaccia semplice e user-friendly che segua il programmatore in ogni fase dello sviluppo.
- Compilatore, un programma dedicato generalmente alla traduzione di un programma sorgente, scritto in codice di alto livello, in un linguaggio binario.
- Command line tool, configurazioni attraverso la quale é possibile fare a meno dell'IDE e seguire lo sviluppo dell'applicazione direttamente da riga di comando, eseguendo particolari procedure in background.

Il risultato della creazione di un'applicazione definita in Symbian C++ é un pacchetto di files, caratterizzati dall'estensione .sis, destinati ad essere installati nel dispositivo.

**Qt.** Dal 2010, Symbian é passato da usare lo standard C++ (Symbian C++) al linguaggio Qt come l'SDK. Qt é una libreria multiplatforma che isola l'utente il piú possibile dalle differenze dei vari sistemi operativi. Servendosi di Qt, l'utente può costruire un'applicazione una volta per tutte, utilizzandola poi su diversi desktop e sistemi operativi. Lo standard seguito da tale linguaggio deriva dal C++: ciò significa che per l'utente che già conosce tale sintassi é sufficiente approfondire pochi dettagli per poter sfruttare questa potente multiplatforma; può essere sviluppato sia con Carbide C++ che con Qt Creator. I dispositivi Symbian possono anche essere programmati usando i linguaggi: Python, Java ME, Flash Lite, Ruby, .NET, Web Runtime Environment (WRT) e Standard C/C++.



## 2 - Progetto

Ogni attività commerciale necessita al giorno d'oggi di ricevere maggiore attenzione da parte della clientela per poter prevalere sui sempre più numerosi concorrenti. È indispensabile quindi una campagna pubblicitaria capillare che tocchi il maggior numero di media possibili quali giornali, internet, televisione ecc. Fornendo una applicazione per dispositivi mobili, inoltre, si facilitano le richieste degli utenti e ci si propone sul mercato con un'immagine moderna e tecnologica. Un'app è l'ideale per fare conoscere il proprio marchio o i servizi offerti ed è uno strumento utilissimo per usufruirne. L'idea di crearne una dedicata a sviluppare il turismo, e in particolare gli hotel, è nata a causa della mancanza su Play Store di applicativi che rappresentassero singoli alberghi. Numerosi sono i motori di ricerca che offrono una serie di soluzioni impostando dei parametri, selezionando le diverse offerte tra vari hotel, ma nessuno appunto è specifico per la singola attività. Lo sviluppo dell'applicazione è stato pensato tenendo in considerazione che potrebbe essere facilmente modificata per poter presentare alberghi di una stessa catena o altri alberghi semplicemente modificandone l'interfaccia. Diverse funzionalità base sono rese disponibili per poter dare un'idea di cosa ogni hotel dovrebbe offrire tramite la sua applicazione rappresentativa.

### Le funzionalità

- Possibilità di visualizzare una galleria di foto riguardanti l'hotel
- Possibilità di condividere foto della galleria direttamente sulle applicazioni social installate (Facebook, Twitter, WhatsApp...) o via e-mail
- Visualizzazione, tramite Google Maps, dell'itinerario da seguire per arrivare all'albergo partendo dall'attuale locazione
- Visualizzazione di tutte le offerte disponibili e del relativo dettaglio
- Possibilità di prenotare la stanza

Nel resto del capitolo si andranno ad analizzare nel dettaglio le diverse funzionalità messe a disposizione dall'applicazione.

## 2.1 – Progettazione applicazione

Di seguito verranno visualizzate una ad una le singole schermate e le relative funzionalità che appaiono nell'applicazione.

### 2.1.1 - Splash Screen



*Figura 3 - Splash screen*

La prima schermata che appare all'apertura dell'applicazione è quella della Splash Activity. Una Splash Activity è un'immagine o un'animazione che viene mostrata all'utente in attesa che l'applicativo esegua operazioni di inizializzazione o di caricamento. Se viene scelta un'immagine piuttosto che un'animazione è buona norma sovrapporre una scritta o una barra di progresso le quali mostrano all'utente che sono in atto operazioni indipendentemente dalla schermata fissa, al fine di evitare che venga arrestata l'applicazione pensando che si sia bloccata. In altre parole essi forniscono all'utente un feedback che un lungo processo è in corso. La Splash Activity viene fatta scomparire al termine delle operazioni di inizializzazione e di caricamento per lasciare posto alla schermata Home. Una Splash Activity è inoltre un buon modo per dare un'ottima impressione grafica, pertanto va creata rispettando lo stile e i colori dell'applicazione e dell'icona di lancio.

## 2.1.2 - Home



*Figura 4 - Home*

La schermata Home é la prima schermata fissa dalla quale é possibile interagire con l'applicazione. Da questa Activity è possibile raggiungere tutte le funzionalità ed é inoltre una delle tre principali suddivisioni dell'applicazione assieme ad Offerte ed a Last Minute. Essa appare con il nome dell'hotel in alto con accanto il bottone per accedere alla mappa, sotto, al centro, sono presenti le informazioni come la via, il telefono, il fax e l'e-mail. Ancora più sotto compare il bottone per accedere alla Gallery delle foto e dai tasti posti a fondo pagina si possono raggiungere le categorie sopra citate.

### 2.1.3 - Griglia e Gallery

Accedibile tramite il pulsante “Hotel”, la Griglia mostra all’utente una anteprima di tutte le immagini, che riguardano l’hotel, presenti nell’applicazione.



*Figura 5 - Griglia foto*



*Figura 6 – Foto a schermo intero*



*Figura 7 – Condivisione Foto*

È possibile aprire le immagini a schermo intero cliccandole e scorrere lateralmente con la gesture di slide per visualizzare quelle precedenti e quelle successive.

## 2.1.4 - Mappa

Da qui è possibile ricevere informazioni riguardanti la locazione dell'hotel.

La funzione utilizza Google Maps ed essendo un servizio web è necessario quindi anche possedere una connessione alla rete per poter inviare la query di richiesta ai server di Google. La risposta verrà poi elaborata dall'applicazione per poter disegnare l'itinerario.

Alla sezione vi si accede cliccando sul segnalino presente nella home. Nel caso in cui sul dispositivo mobile non sia attivato il modulo GPS apparirà una finestra di dialogo che chiederà all'utente se vuole accedere alle impostazioni di sistema.



*Figura 8 - Attivazione GPS*



*Figura 9 - Impostazioni di localizzazione*



*Figura 10 - Termini di utilizzo*

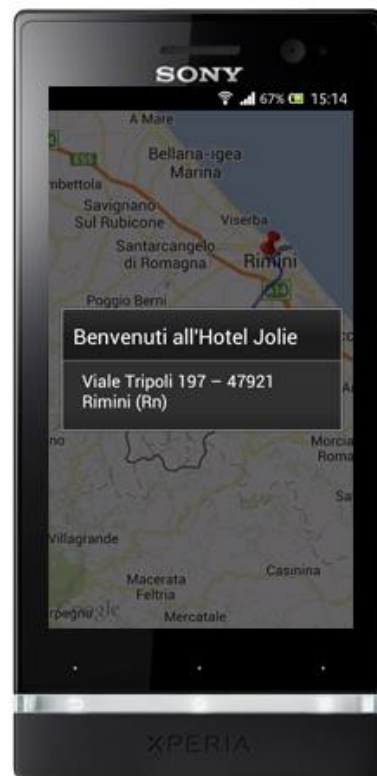
Accettando le condizioni e tornando alla mappa ci si accorgerà che verrà disegnato sulla mappa automaticamente l'itinerario più breve per raggiungere l'hotel partendo dalla posizione attuale. Tappando il segnalino che mostra la nostra posizione apparirà una finestra di dialogo che ci confermerà che ci troviamo realmente in quel punto. Tappando l'altro segnalino apparirà il nome e l'indirizzo dell'hotel destinazione. Se si è in movimento ogni volta che si aprirà la mappa il percorso verrà ricalcolato.



*Figura 11 - Itinerario per l'hotel*



*Figura 12 - Messaggio alla partenza*



*Figura 13 - Messaggio all'arrivo*

Nel caso non si voglia rendere nota la propria posizione (scegliendo No nella finestra di dialogo) verrà visualizzata la sola locazione dell'albergo con il rispettivo segnalino.

## 2.1.5 - Offerte



*Figura 14 - Lista delle offerte*

La sezione Offerte si raggiunge tappando l'apposito bottone posto in fondo ad ognuna delle tre Activity principali. Essa offre una panoramica sulle offerte disponibili in quel preciso momento, elencandole e specificandone il periodo, il tipo di camera e il prezzo.

## 2.1.6 - Dettaglio Offerta



*Figura 15 - Dettaglio offerta*

Scegliendo poi una determinata offerta si avrà la possibilità di visualizzare la descrizione dettagliata semplicemente selezionandola tra quelle proposte. Una nuova Activity verrà lanciata facendo apparire una foto della stanza, le peculiarità e i servizi messi a disposizione dall'offerta stessa. Sarà inoltre possibile prenotarla utilizzando il bottone "Prenota l'offerta".



## 2.1.7 - Prenota

The image shows a black Sony Xperia smartphone. The screen displays a red background with the title 'Prenotazione' at the top. Below the title are four white input fields, each with a label to its left: 'Nome', 'Cognome', 'Telefono', and 'Email'. At the bottom of the form is a white button with the text 'Invia' in red. The status bar at the top of the phone shows 'SONY', signal strength, 60% battery, and the time 16:09. The 'XPERIA' logo is visible at the bottom of the phone's bezel.

*Figura 16 – Form per la prenotazione*

Questa Activity contiene una form che richiederà i dati all'utente interessato alla prenotazione in modo da poter riservare la camera nel periodo indicato dall'offerta. Precisamente i vari campi richiedono: nome, cognome, telefono, email. L'amministratore che ha inserito l'offerta vedrà, oltre ai dati spediti dall'utente, anche le informazioni riguardanti il periodo di soggiorno, il numero di stanza, il costo e l'ID del device che ha effettuato la prenotazione, tutti quanti sulla stessa linea nella tabella delle prenotazioni. Questi ultimi dati saranno già presenti nel server dal momento in cui verrà resa disponibile l'offerta dall'applicazione.

## 2.1.8 - Last Minute



*Figura 17 - Last minute*

Come l'Activity Offerte mostra un'anteprima delle proposte di alloggio ed è anch'essa raggiungibile tramite un bottone a fondo di ogni pagina.

A differenza di Offerte però la scheda Last Minute indica che quelli visualizzati sono gli ultimi posti disponibili e quindi i prezzi saranno notevolmente più convenienti.

## **2.2 - Progettazione server**

La parte server è necessaria per comunicare all'applicazione le eventuali offerte disponibili e per immagazzinare le relative prenotazioni. In particolare la comunicazione tra le due parti avviene tramite una richiesta post http al server che risponderà allo stesso modo sempre con un file JSON strutturato in modo da contenere le informazioni suddivise per tipologia.

### **2.2.1 - Struttura JSON**

Il formato JSON (JavaScript Object Notation) è basato sul linguaggio JavaScript ed è utilizzato per memorizzare i vari tipi di dati utilizzati nei linguaggi di programmazione, supporta fra gli altri booleani, numeri, stringhe, array, array associativi.

Si è scelto l'utilizzo del formato JSON perchè adatto all'ambito mobile grazie alla leggerezza che ne velocizza la trasmissione e alla semplicità con cui è possibile leggere e modificare un file che utilizzi questo formato.

## 2.3 - Progettazione sito web per amministratore

È stata creata un'interfaccia molto semplice e veloce per permettere agli amministratori, a chi crea le offerte e a chi prende le prenotazioni di svolgere basilari operazioni.

### 2.3.1 – Crea nuova offerta

Grazie a questa interfaccia è possibile aggiungere alla lista delle offerte disponibili nel database un ulteriore tupla solamente dopo avere verificato l'identità dell'amministratore tramite la password.

#### Crea nuova offerta

admin password

stanza

data\_in

data\_out

descrizione

prezzo  euro

*Figura 18 - Crea nuova offerta*

### 2.3.2 – Visualizza Prenotazioni

Un'altra pagina permette di mostrare in una tabella le prenotazioni effettuate. La tabella presenta le tuple della tabella Offerte che sono state riservate da un cliente in un periodo successivo al giorno della consultazione, non vengono visualizzate quindi prenotazioni passate.

#### Prenotazioni

Periodo	Stanza	Prezzo	Cliente	Telefono	email	Device ID
da 2015-11-02 a 2015-11-25	4	230	Pippo Spagni	3239848299	pippo.spagni@gmail.com	34j3h332j3jeke617u5i3k

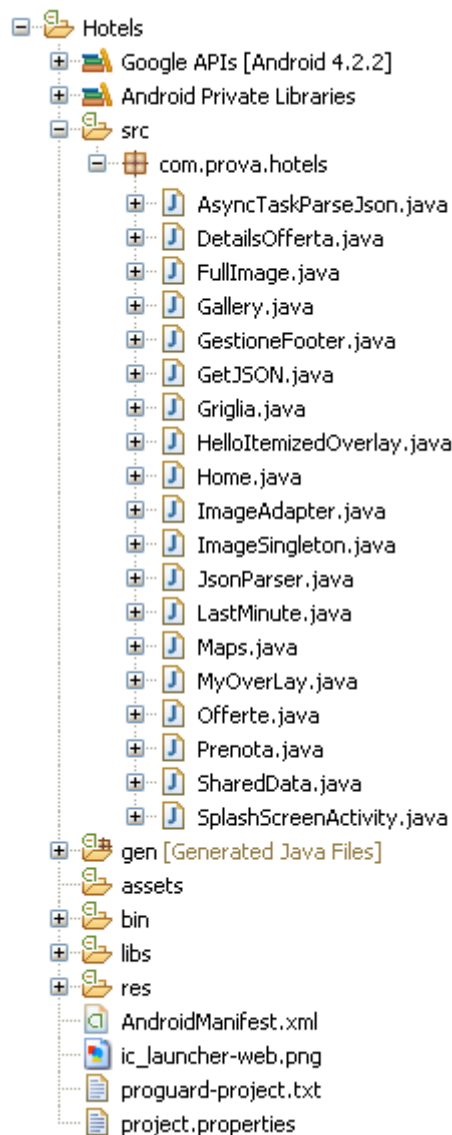
*Figura 19 - Visualizza prenotazioni*

## 3 - Implementazione della soluzione su Android

In questo capitolo vengono presentati i dettagli implementativi delle funzionalità principali descritte precedentemente.

### 3.1 – Implementazione applicazione

Vengono presentate di seguito le singole classi che compongono l'applicazione mobile e ne vengono spiegate le principali procedure.



*Figura 20 - File di progetto*

### 3.1.1 - Classe SplashScreenActivity

La classe SplashScreenActivity permette all'applicazione di svolgere le attività di configurazione iniziale in background mentre all'utente viene mostrata una schermata con il logo dell'hotel e una progress bar.

```
public class SplashScreenActivity extends Activity {
    public ProgressDialog progressDialog;
    private ImageView splashImage;
    public Context context;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        context = SplashScreenActivity.this;

        splashImage = (ImageView) findViewById(R.id.imageView);
        splashImage.setImageResource(R.drawable.splash_screen_finale);

        new BackgroundAsyncTask().execute();
    }
}
```

Figura 21 - Classe SplashScreenActivity

Al lancio dell'activity vengono associate le relative immagini e layout e viene creata una nuova istanza di *BackgroundAsyncTask*.

```
public class BackgroundAsyncTask extends AsyncTask <Void, Void, Void> {

    //dichiarazione dell'oggetto JSONparser
    public JSONParser json;
}
```

Figura 22 - Classe BackgroundAsyncTask

Qui viene dichiarato un oggetto *JSONParser* che ci servirà per la comunicazione con il database. All'interno della classe inoltre si trovano tre metodi.

```
@Override
protected void onPreExecute() {
    progressDialog = ProgressDialog.show(context, getString(R.string.progress_dialog_title),
        getString(R.string.progress_dialog_message));
    progressDialog.setIcon(R.drawable.use_icon);
}
}
```

Figura 23 - Metodo onPreExecute

Il metodo *onPreExecute* prepara l'interfaccia che visualizzerà l'utente.

```

@Override
protected Void doInBackground(Void... params) {
    //Inizializzazione dell'oggetto JSON chiamando il costruttore della classe JSONParser
    json = new JSONParser(context);
    return null;
}

```

*Figura 24 - Metodo doInBackground*

Il metodo *doInBackground* istanzia l'oggetto JSONParser precedentemente dichiarato.

```

@Override
protected void onPostExecute(Void result) {
    Intent intent = new Intent(context, MainActivity.class);
    startActivity(intent);
}

```

*Figura 25 - Metodo onPostExecute*

Infine il metodo *onPostExecute* lancia l'activity principale che sarà quella dalla quale sarà possibile cominciare ad utilizzare l'applicazione.

### 3.1.2 - Classe Home

La classe Home è il punto di partenza per poter usufruire di tutte le funzionalità offerte dall'applicazione tramite degli Intent verso le altre Activity.

```

public class Home extends GestioneFooter {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_home, menu);
        return true;
    }

    //Chiamata quando l'utente tappa il bottone Mappa
    public void openMaps(View view) {
        //Apre la mappa
        Intent intent = new Intent(this, Maps.class);
        startActivity(intent);
    }

    // Chiamata quando l'utente tappa il bottone Hotel
    public void openGallery(View view) {
        Intent intent = new Intent(this, Griglia.class);
        startActivity(intent);
    }
}

```

*Figura 26 - Classe Home*

### 3.1.3 - Classe GestioneFooter

Anche la classe GestioneFooter tramite degli Intent permette di raggiungere le tre Activity principali: Home, Offerte e Last Minute. Queste tre classi estendono proprio GestioneFooter in modo da avere un'interfaccia identica per le tre schermate.

```
public class GestioneFooter extends Activity{

    // Chiamata quando l'utente tappa il bottone Home
    public void sendHome(View view) {
        Intent intent = new Intent (this, Home.class);
        startActivity(intent);
    }

    // Chiamata quando l'utente tappa il bottone Offerte
    public void sendOfferte(View view) {
        Intent intent = new Intent (this, Offerte.class);
        startActivity(intent);
    }

    // Chiamata quando l'utente tappa il bottone Last Minute
    public void sendLastMinute(View view) {
        Intent intent = new Intent (this, LastMinute.class);
        startActivity(intent);
    }
}
```

*Figura 27 - Classe GestioneFooter*



### 3.1.4 - Classe Griglia

La classe Griglia da la possibilità di visualizzare le immagini presenti nell'array nella classe ImageSingleton, in una vista a scacchiera in formato anteprima. Le stesse saranno quindi ridimensionate per poter selezionare quale aprire a schermo intero tra quelle presenti.

```
public class Griglia extends Activity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_griglia);

        //Collego l'istanza alla GridView effettiva
        GridView gridView = (GridView) findViewById(R.id.gridview);

        //Collego l'Adapter all'istanza di GridView
        gridView.setAdapter(new ImageAdapter(this));

        //Per ogni elemento della GridView
        gridView.setOnItemClickListener(new OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
                //Manda l'immagine all'activity Gallery
                Intent i = new Intent(getApplicationContext(), Gallery.class);
                //Gli passa l'indice dell'array
                i.putExtra("id", position);
                startActivity(i);
            }
        });
    }
}
```

*Figura 28 - Classe Griglia*

Il metodo *onItemClick* indica all'Activity Gallery quale immagine dovrà essere visualizzata distinguendola dalle altre tramite la posizione nell'array.

### 3.1.5 - Classe Gallery

La classe Gallery consente di presentare le immagini presenti nell'array della classe ImageSingleton, nell'omonima Activity. Esse saranno mostrate a schermo intero, con un formato adattato rispetto alle dimensioni reali delle immagini, in modo da poterle avere tutte della stessa grandezza.

Nel corso di questa trattazione si andranno a descrivere alcune delle principali funzionalità messe a disposizione da questa classe.

```
public class Gallery extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gallery);
        ViewPager viewPager = (ViewPager) findViewById(R.id.view_pager);
        ImagePagerAdapter adapter = new ImagePagerAdapter();
        viewPager.setAdapter(adapter);
    }
}
```

*Figura 29 - Classe Gallery*

### 3.1.6 - Classe Maps

La classe Maps comprende una serie di metodi che permettono di visualizzare sulla mappa l'itinerario per raggiungere dalla posizione attuale quella dell'hotel. È necessario quindi che il dispositivo abbia un modulo GPS attivo.

```
public class Maps extends MapActivity {
    /** Called when the activity is first created. */
    MapView mapView;
    private Maps _activity;
    GeoPoint srcGeoPoint, destGeoPoint;
    private static List<Overlay> mOverlays;
```

*Figura 30 - Classe Maps*

Inizialmente vengono dichiarate le variabili necessarie per la localizzazione.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mapView = new MapView(this, "ARNGrZz8OR4rsZ6dq67cVpukA");
    mapView.setClickable(true);
    setContentView(mapView);

    _activity = this;
    LocationManager lm;

    double src_lat = 43.972468;
    double src_long = 12.489655;
    double dest_lat = 44.057082;
    double dest_long = 12.5646;
```

*Figura 31 - Metodo onCreate*

Il metodo *onCreate()* è il primo ad essere eseguito. All'interno di esso viene associato il rispettivo layout con il codice identificativo fornito da Google e vengono stabilite le coordinate dei due punti da rilevare sulla mappa.

```

//Tramite la risposta bool del metodo isProviderEnabled vediamo il suo stato
boolean enabled = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
//Se non è abilitato
if (!enabled){
    //Crea una AlertDialog dove chiede se lo si vuole abilitare
    AlertDialog.Builder gpsDialog = new AlertDialog.Builder(this);
    gpsDialog.setTitle("GPS disabilitato");
    gpsDialog.setMessage("Vuoi attivarlo dalle impostazioni?");

    //Non permette di fare BACK una volta apparsa la dialog
    gpsDialog.setCancelable(false);

    //Creazione del bottone SI
    gpsDialog.setPositiveButton("SI", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            //Apri le impostazioni di localizzazione
            Intent intent = new Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
            startActivity(intent);
        }
    });

    //Creazione del bottone NO
    gpsDialog.setNegativeButton("NO", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            //torna alla mappa senza fare nulla
        }
    });

    //Crea di fatto la AlertDialog e la mostra a video
    AlertDialog alert = gpsDialog.create();
    alert.show();
}

```

**Figura 32 - Metodo onCreate parte 2**

Sempre all'interno di *onCreate()* viene fatto un controllo per verificare se è attivo il modulo GPS del dispositivo. In caso contrario viene chiesto all'utente, tramite una *AlertDialog*, se intende attivarlo: nel caso di risposta negativa verrà visualizzata la mappa con solamente un segnalino sulla destinazione; nel caso di risposta affermativa verrà aperta la pagina delle impostazioni di localizzazione di sistema.

### 3.1.7 - Classe MyOverlay

La classe MyOverlay consente di avere sopra l'effettiva mappa fornita da Google Maps un layout personalizzabile. Quello che ci interessa per l'applicazione è poter disegnare un percorso in base alle coordinate di partenza e di arrivo inviate.

```
public class MyOverLay extends Overlay
{
    private GeoPoint gp1;
    private GeoPoint gp2;
    private int mode=0;
    private int defaultColor;
    Context mContext;

    public MyOverLay(Context context,GeoPoint gp1,GeoPoint gp2,int mode)
    {
        this.gp1 = gp1;
        this.gp2 = gp2;
        this.mode = mode;
        this.mContext = context;
        defaultColor = 999; // no defaultColor
    }

    public MyOverLay(GeoPoint gp1,GeoPoint gp2,int mode, int defaultColor)
    {
        this.gp1 = gp1;
        this.gp2 = gp2;
        this.mode = mode;
        this.defaultColor = defaultColor;
    }
    public int getMode()
    {
        return mode;
    }
}
```

*Figura 33 - Classe MyOverlay*

Il metodo *draw* disegna effettivamente l'itinerario e permette di impostare le varie caratteristiche degli oggetti da sovrapporre alla *mapView*.

```
@Override
public boolean draw
(Canvas canvas, MapView mapView, boolean shadow, long when)
{
    Projection projection = mapView.getProjection();
    if (shadow == false)
    {
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        Point point = new Point();
        projection.toPixels(gp1, point);

        if(mode==1)
        {
            if(defaultColor==999)
                paint.setColor(Color.BLUE);
            else
                paint.setColor(defaultColor);
        }
        else if(mode==2)
        {
            if(defaultColor==999)
                paint.setColor(Color.RED);
            else
                paint.setColor(defaultColor);
            Point point2 = new Point();
            projection.toPixels(gp2, point2);
            paint.setStrokeWidth(5);
            paint.setAlpha(120);
            canvas.drawLine(point.x, point.y, point2.x,point2.y, paint);
        }
        else if(mode==3)
        {
            if(defaultColor==999)
                paint.setColor(Color.GREEN);
            else
                paint.setColor(defaultColor);
            Point point2 = new Point();
            projection.toPixels(gp2, point2);
            paint.setStrokeWidth(5);
            paint.setAlpha(120);
            canvas.drawLine(point.x, point.y, point2.x,point2.y, paint);
        }
    }
    return super.draw(canvas, mapView, shadow, when);
}
```

*Figura 34 - Metodo draw*

### 3.1.8 - Classe JsonParser

Questa classe contiene un metodo che permette, passandogli un indirizzo URL, di far comunicare l'applicazione con il database.

```
public class JsonParser {

    final String TAG = "JsonParser.java";

    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    public JSONObject getJSONFromUrl(String url) {

        // make HTTP request
        try {

            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        try {

            BufferedReader reader = new BufferedReader(new InputStreamReader(is, "iso-8859-1"), 8);
            StringBuilder sb = new StringBuilder();
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();
            json = sb.toString();

        } catch (Exception e) {
            Log.e(TAG, "Error converting result " + e.toString());
        }

        // try parse the string to a JSON object
        try {
            jsonObj = new JSONObject(json);
        } catch (JSONException e) {
            Log.e(TAG, "Error parsing data " + e.toString());
        }

        // return JSON String
        return jsonObj;

    }
}
```

*Figura 35 - Classe JsonParser*

Viene fatta richiesta al determinato URL che restituirà un file in formato JSON il quale verrà letto e convertito in stringa. Solo successivamente verrà riconvertito in un oggetto JSON.

## 3.2 - Implementazione database

La realizzazione della parte server è avvenuta utilizzando l'editor notepad++ e il linguaggio di programmazione php.

Per poter rendere disponibili le funzionalità descritte nel precedente capitolo, è necessario dotare l'applicazione di una struttura dati in cui possa salvare e da cui possa estrarre informazioni riguardanti le prenotazioni e le offerte disponibili. È stato perciò progettato un piccolo database su AlterVista formato da una sola tabella dove ogni tupla si riferisce ad una singola offerta. Originariamente, all'inserimento dell'offerta da parte dell'amministratore, saranno presenti solo i campi *id* che è l'identificativo dell'offerta, *stanza* che è il numero della stanza, *data\_in* il giorno dal quale l'offerta ha inizio, *data\_out* il giorno nella quale l'offerta finisce, *descrizione* una breve spiegazione dei servizi proposti e *prezzo* il costo fissato. In seguito ad una prenotazione invece verranno riempiti anche *nome*, *cognome*, *telefono*, *email* e *device\_id* con i dati dell'utente che l'ha effettuata. La presenza di un valore non nullo in questi campi indicherà che l'offerta non è più disponibile.

L'applicazione, per ricevere la lista delle offerte disponibili aggiornata, domanderà alla pagina “offerte.php” un file in formato JSON contenente l'informazione necessaria.

Di seguito viene mostrato il codice di “**offerte.php**” :

```
<?php
include 'appConfig.php';
dbStartConnection();
$sql = 'SELECT * FROM offerta WHERE data_in >= now() AND nome = ""';
$sqlCheck = mysql_query($sql);
$array = array();
while($res = mysql_fetch_assoc($sqlCheck))
    $array[ ] = $res;
echo json_encode($array);
?>
```

Il codice permette di creare un array con tutte le offerte disponibili, quindi quelle non scadute senza prenotazione, e metterlo all'interno di un file JSON.

Qualora l'utente, una volta visionate le disponibilità, decida di prenotare una stanza i suoi dati verranno inviati alla pagina “**prenota.php**” di cui mostriamo il codice:



```

<?php
include 'appConfig.php';
dbStartConnection();
$message = 'Errore interno.';
$id = strip_tags(mysql_real_escape_string($_POST["id"]));
$nome = strip_tags(mysql_real_escape_string($_POST["nome"]));
$cognome = strip_tags(mysql_real_escape_string($_POST["cognome"]));
$email = strip_tags(mysql_real_escape_string($_POST["email"]));
$telefono = strip_tags(mysql_real_escape_string($_POST["telefono"]));
$deviceId = strip_tags(mysql_real_escape_string($_POST["device_id"]));

if ($id != '' && $nome != '' && $cognome != '' &&
    $email != '' && $telefono != '' && $deviceId != '') {
    $q1 = ''.'SELECT id ''FROM offerta ''WHERE id = '. $id .'
        ''AND nome = "" '. ''AND data_in >= now()';
    $q1Check = mysql_query($q1);

    if(mysql_num_rows($q1Check) == 0) {
        $message = 'Questa offerta non è più disponibile';
    }
    else {
        $q2 = ''.'UPDATE offerta '. ''SET nome = "".$nome.'",
            cognome = "".$cognome.'", email = "".$email.'",
            telefono = "".$telefono.'", device_id =
                "".$deviceId.'" '.
            'WHERE id = '. $id;
        $q2Check = mysql_query($q2);
        $message = '1';
    }
}
else {
    $message = 'Tutti i campi devono essere riempiti.';
}
echo $message;
?>

```

La pagina controlla che l'utente abbia inserito una voce in tutti i campi al momento della prenotazione, se è così verrà aggiornata l'offerta stessa con i dati inviati dall'applicazione al server.

### 3.3 - Implementazione sito web per amministratore

Il progetto necessita anche una parte gestionale per l'inserimento di nuove offerte e l'interrogazione delle prenotazioni effettuate. Sono state create due pagine distinte "admin.php" e "prenotazioni.php" che vengono riportate di seguito.

"admin.php" :

```
<?php
include 'appConfig.php';
dbStartConnection();
$message = 'Errore interno.';
$password = strip_tags(mysql_real_escape_string($_POST["password"]));
$stanza = strip_tags(mysql_real_escape_string($_POST["stanza"]));
$dataIn = strip_tags(mysql_real_escape_string($_POST["data_in"]));
$dataOut = strip_tags(mysql_real_escape_string($_POST["data_out"]));
$descrizione = strip_tags(mysql_real_escape_string($_POST["desc"]));
$prezzo = strip_tags(mysql_real_escape_string($_POST["prezzo"]));

if ($password != '' && $stanza != '' && $dataIn != '' && dataOut != ''
&& $descrizione != '' && $prezzo != '' && $password != '') {
    if ($password == ADMIN_PASSWORD) {
        $q1 = 'INSERT INTO offerta (stanza, data_in, data_out,
desc,prezzo)'. 'VALUES("'. $stanza. '",'. $dataIn. '", "'. $data
Out. '", "'. $descrizione. '", "'. $prezzo. '")';
        $q1Check = mysql_query($q1);
        $message = 'Offerta inserita!';
    }
    else {
        $message = 'Password errata!';
    }
}
else {
```

```

    if ($password == '' && $stanza == '' && $dataIn == '' &&
        $dataOut == '' && $descrizione == '' && $prezzo == '')
        $message = '';
    else
        $message = 'Tutti i campi devono essere riempiti.';
}
echo '<hr>';
if ($message != '')
    echo $message.'<hr>';
?>

<h3>Crea nuova offerta</h3>
<form name="input" action="admin.php" method="post">
    admin password <input type="text" name="password"><br><br>
    stanza <input type="text" name="stanza"><br>
    data_in <input type="text" name="data_in"><br>
    data_out <input type="text" name="data_out"><br>
    desc <textarea rows="5" cols="40" name="descrizione"></textarea><br>
    prezzo <input type="text" name="prezzo">euro<br>
    <input type="submit" value="Submit">
</form>
<br>
<hr>

```

Questa pagina consente ad un amministratore di aggiungere nuove offerte alla tabella. Tutti i campi della form devono essere riempiti e, per poter effettuare l'accesso, la password deve coincidere con quella registrata.

**“prenotazioni.php” :**

```
<html>
<body>
<hr>
<?php
include 'appConfig.php';
dbStartConnection();
$q1 = 'SELECT stanza, data_in, data_out, prezzo, nome, cognome,
telefono, device_id, email FROM offerta WHERE data_in >= now() AND
nome != ""';
$q1Check = mysql_query($q1);
echo '<table border="1">';
echo '<tr>';
echo '<th>Periodo</th>';
echo '<th>Stanza</th>';
echo '<th>Prezzo</th>';
echo '<th>Cliente</th>';
echo '<th>Telefono</th>';
echo '<th>email</th>';
echo '<th>Device ID</th>';
echo '</tr>';
while($res = mysql_fetch_assoc($q1Check)) {
    echo '<tr>';
    echo '<td> da '.$res['data_in'].' a '.$res['data_out'].'</td>';
    echo '<td> '.$res['stanza'].' </td>';
    echo '<td> '.$res['prezzo'].' </td>';
    echo '<td> '.$res['nome'].' '.$res['cognome'].' </td>';
    echo '<td> '.$res['telefono'].' </td>';
    echo '<td> '.$res['email'].' </td>';
    echo '<td> '.$res['device_id'].' </td>';
    echo '</tr>'; }
echo '</table>';
?>
<hr>
</body>
</html>
```

Il codice permette di visualizzare in una tabella le prenotazioni fino a quel momento effettuate. Dal database vengono selezionate solo le offerte che sono successive alla data dell'interrogazione e che hanno il campo "nome" non vuoto, ciò significa che solo quelle che sono state riservate saranno mostrate.



## **4 – Sviluppi futuri**

L'applicazione presentata potrebbe subire ulteriori importanti miglioramenti, così come potrebbe prevedere l'aggiunta di nuove funzionalità.

Lo sviluppo per dispositivi Apple potrebbe essere il prossimo passo ad essere intrapreso in modo da rendere disponibile l'applicazione a un'ulteriore fetta di mercato.

Un'altra direzione in cui sarebbe utile far crescere l'applicazione è sicuramente l'aspetto grafico, molto semplice e basilare attualmente. L'aggiunta di migliorie porterebbe a una più piacevole e immediata consultazione.

Inoltre potrebbe essere offerta la possibilità agli utenti di lasciare recensioni dell'hotel che siano consultabili da tutti in modo di dare la possibilità a chi scarica l'applicazione, e agli amministratori, di avere un feedback dei servizi proposti.





## **5 – Conclusioni**

Attraverso questa tesi è stato possibile descrivere nel dettaglio la realizzazione di un'applicazione per la presentazione di un generico hotel. Come spiegato anche nel corso della trattazione, lo sviluppo di applicazioni per dispositivi mobile è un'attività che ha acquisito, negli ultimi anni, un rilievo sempre maggiore ed è pertanto una materia di studio interessante che merita di essere approfondita.

Questa esperienza mi ha permesso di capire come creare e organizzare un intero progetto e come distribuire il carico di lavoro affrontando le inevitabili problematiche che si incontrano sviluppandolo. Le varie fasi di analisi, progettazione e sviluppo hanno richiesto, essendo la mia prima applicazione mobile sviluppata, numerose consultazioni di guide e soprattutto di esempi pratici e tutorial.



## 6 – Bibliografia

- [1] [Online] [http://it.wikipedia.org/wiki/Computer\\_desktop](http://it.wikipedia.org/wiki/Computer_desktop)
- [2] [Online] [http://it.wikipedia.org/wiki/Ubiquitous\\_computing](http://it.wikipedia.org/wiki/Ubiquitous_computing)
- [3] [Online] [http://en.wikipedia.org/wiki/Mark\\_Weiser](http://en.wikipedia.org/wiki/Mark_Weiser)
- [4] [Online] <http://www.solotablet.it/tablet-impresa/ambiti-di-applicazione/evoluzione-dei-dispositivi-mobili>
- [5] [Online] <http://www.html.it/pag/15335/i-nuovi-canali-di-accesso-al-web/>
- [6] [Online] <http://it.wikipedia.org/wiki/Smartphone>
- [7] [Online] [http://en.wikipedia.org/wiki/History\\_of\\_mobile\\_phones](http://en.wikipedia.org/wiki/History_of_mobile_phones)
- [8] [Online] [http://it.wikipedia.org/wiki/IPhone\\_EDGE](http://it.wikipedia.org/wiki/IPhone_EDGE)
- [9] [Online] [http://it.wikipedia.org/wiki/Sistema\\_operativo\\_per\\_dispositivi\\_mobili](http://it.wikipedia.org/wiki/Sistema_operativo_per_dispositivi_mobili)
- [10] [Online] [http://it.wikipedia.org/wiki/IOS\\_\(Apple\)](http://it.wikipedia.org/wiki/IOS_(Apple))
- [11] [Online] <http://www.lgblog.it/2010/03/che-cos%E2%80%99e-android-una-guida-facile/>
- [12] [Online] <http://it.wikipedia.org/wiki/Android>
- [13] [Online] [http://it.wikipedia.org/wiki/Windows\\_Phone](http://it.wikipedia.org/wiki/Windows_Phone)
- [14] [Online] [http://it.wikipedia.org/wiki/Microsoft\\_XNA](http://it.wikipedia.org/wiki/Microsoft_XNA)
- [15] [Online] [http://it.wikipedia.org/wiki/BlackBerry\\_OS](http://it.wikipedia.org/wiki/BlackBerry_OS)
- [16] [Online] [http://it.wikipedia.org/wiki/Symbian\\_Foundation](http://it.wikipedia.org/wiki/Symbian_Foundation)
- [17] [Online] [http://it.wikipedia.org/wiki/Symbian\\_OS](http://it.wikipedia.org/wiki/Symbian_OS)