

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

TAKEAWAY

UN'APPLICAZIONE MOBILE PER LA RISTORAZIONE

Relazione finale in
MOBILE WEB DESIGN

Relatore
Dott. Mirko Ravaioli

Presentata da
Mattia Pioggia

Sessione Prima

Anno Accademico 2014/2015

SOMMARIO

Introduzione	7
1.1 L'avanzamento tecnologico	7
1.2 I social network	8
1.3 La Rivoluzione Mobile	9
1.4 Un mondo costantemente connesso	10
Capitolo 1 – Il Progetto	13
2.1 La nascita del progetto	13
2.2 Argomenti trattati	14
2.3 La progettazione.....	15
2.3.1 Progettazione e analisi delle schermate di base.....	16
2.3.2 Progettazione della struttura software	17
2.3.3 Architetture software per un implementazione efficiente	17
2.3.4 Progettazione concettuale della base di dati.....	18
2.3.5 Scelta della piattaforma e del linguaggio del server.....	25
Capitolo 2 - L'implementazione	27
3.1 La scelta del sistema operativo	27
3.1.1 Android	27
3.1.2 iOS	28
3.1.3 Windows Phone	29
3.2 Analisi dell'implementazione (lato Client).....	31
3.2.1 SplashActivity	33
3.2.2 HomeActivity	34
3.2.3 MenuActivity	35
3.2.4 LocaleActivity	36
3.2.4 UtenteActivity	37
3.2.5 Login e Registrazione	38
3.2.6 OrdineActivity.....	39

3.3	Analisi dell'implementazione (lato Server).....	40
3.3.1	Web Hosting.....	41
3.3.2	I servizi	42
3.3.3	Standard per la comunicazione Client/Server.....	43
3.4	Librerie Utilizzate.....	45
3.4.1	Google Play Services (client)	45
3.4.2	Librerie grafiche	46
Capitolo 3 – Conclusione		49
3.1	Sviluppi futuri.....	49
3.2	Sitografia	50

Introduzione

L'obiettivo di questa tesi è quello di descrivere la nascita, la progettazione e l'implementazione di una piattaforma software mobile in grado di gestire ordini TakeAway per conto di un Ristorante. L'applicazione avrà principalmente 2 funzioni; la presentazione al cliente del ristorante, fornendo ogni informazione utile e la possibilità di effettuare e gestire ordinazioni e prenotazioni dai propri dispositivi mobili attraverso un interfaccia grafica semplice ed intuitiva. Il sistema operativo mobile da me scelto per lo sviluppo di questo progetto è stato Android. Questa scelta si è basata in primis sulla diffusione di Android rispetto ai noti rivali iOS di Apple e WindowsPhone di Microsoft, in secondo luogo ho ritenuto molto interessante cogliere i vantaggi di un sistema operativo openSource e libero.

1.1 L'avanzamento tecnologico

Quella che stiamo vivendo negli ultimi anni è sicuramente una grande rivoluzione tecnologica. Grazie a tecniche produttive innovative e ad importanti studi scientifici il mercato degli "apparecchi elettronici" è in continua evoluzione, rinnovandosi costantemente con nuove tecnologie che consentono la produzione di dispositivi sempre più piccoli, più potenti e più sostenibili per l'ambiente. Innovazione e grandi investimenti stanno cambiando inoltre la rete comunicativa globale grazie a nuovi brevetti sulla trasmissioni di dati via cavo e su nuovi protocolli e infrastrutture per potenziare le reti mobili. Stiamo conoscendo, anche in Italia, i grandi benefici dei collegamenti in fibra ottica per soluzioni domestiche e di impresa e delle reti LTE per le connessioni mobili. Assistiamo in questo modo, ad una costante crescita e sviluppo delle tecnologie che popolano il mondo Hi-tech, con una grande predilezione verso gli strumenti mobile piuttosto che sulle soluzioni "fisse" alle quali eravamo molto più legati nel decennio precedente. Queste affascinanti trasformazioni tecnologiche che ogni anno propongono nuove soluzioni hardware e software hanno, a mio avviso, un

risvolto ancora più interessante e incisivo, se ad essere analizzata è la mutazione socioculturale che portano a seguito. La vera rivoluzione alla quale stiamo assistendo negli ultimi 5-10 anni non interessa scoperte scientifiche particolarmente innovative, o rivoluzioni produttive di rilievo, quello che è cambiato e sta cambiando è la posizione che la tecnologia assume nella nostra quotidianità. Con questa premessa dobbiamo obbligatoriamente fare riferimento al primo passo importante verso i nuovi paradigmi comunicativi e ai nuovi possibili utilizzi di Internet, i social network.

1.2 I social network

Internet è stato immediatamente percepito come qualcosa che ci avrebbe cambiato. Tutti nel ventennio scorso hanno iniziato a conoscere qualcosa di nuovo che avrebbe cambiato il nostro modo di vivere. Gradualmente è entrato nelle nostre case e ci ha fatto apprezzare tutte le sue potenzialità. La condivisione di file e la consultazione di enciclopedie o genericamente l'informazione digitale sono diventate icona dell'inizio del nuovo millennio, ma un grande passo in avanti è stato possibile grazie alla nascita dell'instant messaging, ovvero la possibilità di inviare e ricevere messaggi in tempi molto veloci, in questo modo le conversazioni testuali si avvicinavano sempre più alle comunicazioni telefoniche e veniva risolto il problema della lentezza che caratterizzava lo scambio di e-mail. Internet è da subito stato un grande contenitore di informazioni, in qualche modo una "Giungla" di dati di ogni tipologia, dall'informazione scientifica ai contenuti ludici e demenziali, da uno strumento pubblicitario per le aziende ad un modo per tenersi in contatto con i parenti lontani, sicuramente nessuno poteva immaginare che i nostri dati personali e le nostre vicende quotidiane potessero diventare oggetti di interesse tale da essere pubblicati su internet. Ovviamente eravamo in errore. 2004, la rivoluzione di internet e del mondo mobile ha inizio qui. La nascita dei social network, tra dibattiti e confusione generale è stata la "ventata di aria fresca" che ha saputo innovare come forse solo Google in precedenza, la scena informatica mondiale. I social network, anche se potremmo benissimo parlare di uno solo tra questi, hanno messo al centro dell'attenzione la

persona, con i propri gusti e le proprie caratteristiche, internet è diventato uno strumento costruito attorno all'utente dove poter ricercare informazioni vicine a noi e alla nostra cerchia di amicizie. E' nato un approccio "user-oriented" dove chiunque ha la possibilità di esprimere se stesso e comunicare con amici attraverso testo e altre risorse multimediali. Infine l'avvento dei social network ha avuto un ruolo fondamentale nella diffusione di internet in fasce di popolazione molto distanti dal mondo di internet.

1.3 La Rivoluzione Mobile

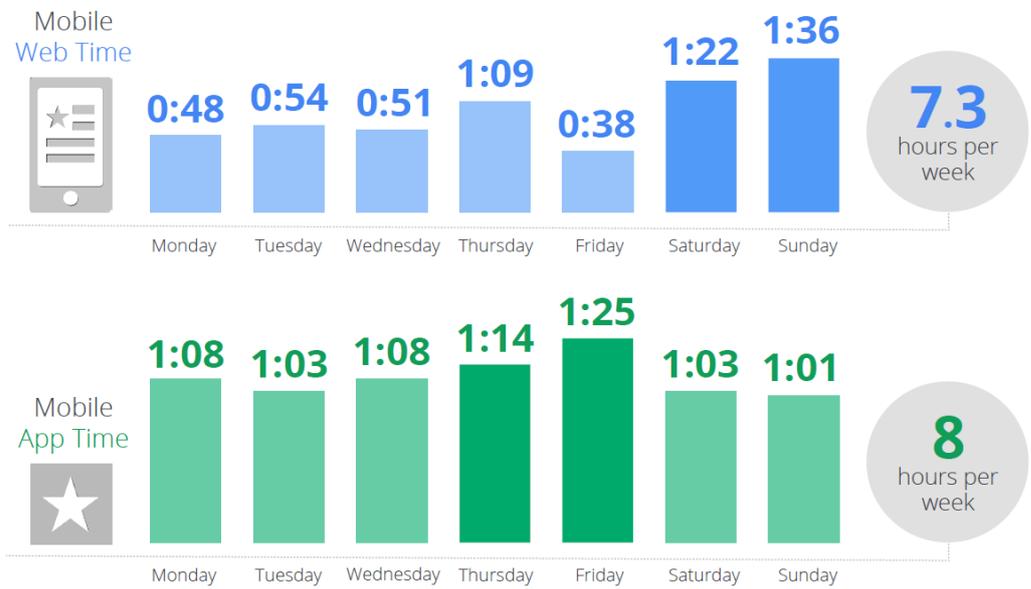
Gli analisti di mercato di tutto il mondo avevano avvisato già da tempo che il mercato dei pc era in pericolo di stallo e avrebbe passato periodi bui, negli ultimi 2 anni la conferma è arrivata. Il settore desktop come noto da tempo ha subito un importante calo delle vendite giustificato solamente dall'aumento esponenziale delle vendite di smartphone, tablet e dispositivi mobili di ogni genere. Le dimensioni e la praticità di utilizzo hanno avuto la meglio sui grandi schermi e sulle grandi prestazioni; non appena le prestazioni offerte da questi dispositivi ha raggiunto una soglia tutto sommato accettabile per i task di tutti i giorni. Perché è avvenuta questa inversione di tendenza? I dispositivi mobili ai quali mi riferisco sono oggetti elettronici con uno schermo touchscreen dalle dimensioni variabili e dalle varie combinazioni di moduli di connettività. Le dimensioni variano in base al form-factor di riferimento, sono generalmente di 4-5 pollici di diagonale per il mondo smartphone mentre superiori ai 7 pollici per i Tablet. Le caratteristiche peculiari di questa categoria di prodotti sono sicuramente il peso contenuto, le dimensioni "comode" e il basso consumo energetico, tralasciando altre funzionalità come la camera o il tracciamento gps... A livello Prestazionale si posizionano sempre al di sotto di alternative desktop e spesso adottano architettura ARM a differenza del mercato desktop che si basa quasi totalmente su soluzioni x86. Questa differenza porta ad avere delle ottime prestazioni in relazione alle dimensioni del prodotto, con consumi decisamente bassi. La caratteristica che ha portato al successo i dispositivi mobili è sicuramente la potenza

adeguata a svolgere qualsiasi esigenza non professionale contenuta in un form-factor tascabile. Un compromesso perfetto che ha fatto rimpiazzare i pc con tablet e smartphone a tutti coloro che utilizzavano il computer per web browsing e poco altro. Una seconda caratteristica che ha determinato la grande diffusione del mondo mobile è da attribuire alle modalità di funzionamento semplici ed intuitive che caratterizzano l'immissione dei dati attraverso l'utilizzo di un display touchscreen, unita alle interfacce applicative user-friendly e standardizzate.

1.4 Un mondo costantemente connesso

“Viviamo in un mondo costantemente connesso“. Inizia così una brillante ricerca di Google in collaborazione con Nielsen, in cui si cerca di analizzare e prevedere il futuro dell'e-Commerce; un futuro che sarà sempre più mobile. Attraverso una raccolta di dati, su un campione di circa mille partecipanti, si è tentato di capire le abitudini dei consumatori dotati di smartphone durante i processi di acquisto e di semplice navigazione web. Tra gli interessanti dati che emergono e ai quali non faccio riferimento si evince che la maggior parte del campione utilizza lo smartphone per più di 15 ore alla settimana e la loro navigazione nella maggioranza dei casi inizia con un motore di ricerca. Questi dati sono un esempio importante nel delineare il fenomeno dei nostri giorni, il futuro è su internet e nel mobile, tanto che iniziano proprio ora ad affacciarsi al mondo mobile anche le piccole aziende localizzate sul territorio che cercano nuovi metodi pubblicitari e comunicativi e delle opportunità per ampliare e ottimizzare i propri servizi. Da questa considerazione nasce l'idea del progetto che sto per presentare. Ovvero un'applicazione mobile destinata a ristoranti e locali che già offrono un servizio di asporto o di consegna di prodotti a domicilio, che siano interessati a pubblicizzare il proprio locale su dispositivi mobili e che vogliano informatizzare e semplificare il sistema di ordini e consegne dei prodotti sul territorio.

Consumers spend more than 15 hours per week on mobile research



[Figura 1 – Tempo speso in ricerche]

Capitolo 1 – Il Progetto

Il progetto che verrà descritto è nato in stato embrionale come progetto di tirocinio, evolutosi successivamente in un progetto completo a 360 gradi che affronta tutte le problematiche relative all'utilizzo di strumenti diversi volti alla perfetta cooperazione per una esperienza utente completa, facile e accattivante. Il progetto è basato sul corso di studi "Mobile Web Design" del professore Mirko Ravaioli e affronta tutte le difficoltà e le potenzialità del mondo Mobile, unite alle importanti possibilità offerte dal Web. Il progetto descrive le potenzialità comunicative e sociali della più recente offerta tecnologia.



[Figura 2 – Logo Applicazione TakeAway]

2.1 La nascita del progetto

L'idea alla base dell'applicazione progettata nasce dalla necessità di rinnovamento di una categoria di imprenditori che cerca innovazione in un mercato che da tempo è produttivo, stabile, ma troppo omologato. La continua nascita di ristoranti e pizzerie che offrono gli stessi servizi di asporto e consegna a domicilio genera una sempre maggiore concorrenza, la quale può essere combattuta solamente con un grande lavoro di ottimizzazione dei costi e, soprattutto di tempo. Sono proprio la gestione

del tempo e l'organizzazione del lavoro lo scoglio maggiore in questo settore, ed ogni fallimento e disservizio si ripercuote in primis sul cliente e in seconda battuta sull'imprenditore. A questo si aggiunge la continua ricerca da parte dei clienti di piattaforme software mobile, in grado di semplificare le operazioni di tutti i giorni, risparmiando tempo e migliorando un servizio, altrimenti più approssimativo e meno sicuro. Valutando queste due necessità del "settore" è stata colta l'occasione per sviluppare un servizio che nel nostro paese non si è ancora diffuso, cercando di creare uno strumento che avesse una doppia utilità, ovvero la presentazione del locale, con tutte le informazioni e la possibilità di effettuare ordinazioni online attraverso il proprio smartphone.

2.2 Argomenti trattati

Come anticipato precedentemente l'applicazione TakeAway sarà utilizzata dagli utenti per cogliere tutte le informazioni relative ad un ristorante che conosciamo o di cui abbiamo sentito parlare attraverso due moduli, ovvero il "Menu" che presenta al cliente le pietanze offerte con relative fotografie, prezzi e accurate descrizioni, come se stessimo sfogliando il menu cartaceo del locale ed il modulo "Locale" che presenta il ristorante con fotografie degli ambienti, informazioni di contatto e informazioni sui servizi offerti. E' presente inoltre all'interno di questo modulo, la possibilità di inviare messaggi, email o chiamate direttamente dall'app attraverso icone intuitive. Queste due sezioni dell'applicazione non necessitano di alcun login e funzionano anche in modalità offline (con alcune piccole limitazioni date dall'impossibilità di sincronizzare le informazioni con il server). Il fulcro dell'applicazione ruota invece attorno alla possibilità di effettuare ordini online per mezzo del proprio smartphone o tablet.

L'applicazione consente di effettuare un ordine scegliendo la modalità desiderata tra consegna a domicilio e asporto, con conseguente ritiro al ristorante, nel secondo caso. Durante la creazione dell'ordine attraverso il modulo "ordina", si può consultare il menu selezionando le pietanze desiderate e dopo aver scelto l'orario ed altre

informazioni, quali eventuali commenti per il cuoco o la modifica dell'indirizzo, si passa alla selezione dell'orario di consegna, a questo punto la conferma dell'ordine registrerà l'ordinazione sul database del ristorante. L'utilizzo di questo modulo e del modulo "utente" che descriverò a breve è possibile solo in presenza di connessione ad internet; non è d'obbligo, invece una registrazione, per agevolare l'utilizzo veloce a chi non desidera registrarsi al servizio, ma vuole limitarsi ad un accesso da Ospite. Il modulo "utente" prima citato, rappresenta il pannello di controllo dell'utente, qui dopo avere eseguito il login (o registrazione) è possibile consultare l'elenco degli ordini effettuati, oppure visionare e modificare i dati utente. L'utilizzo di questa funzionalità è stato reso possibile anche a coloro che effettuano gli ordini senza essere registrati, ma da ospiti con credenziali temporanee. Le informazioni relative all'utente iscritto sono ridotte al minimo indispensabile, con lo scopo di non annoiare l'utente perdendo tempo nell'iscrizione. I dati richiesti all'iscrizione comprendono anche un recapito telefonico e l'indirizzo principale per velocizzare l'esecuzione dell'ordine, è comunque consentita la modifica dei dati prima dell'invio di ogni ordine, per consentire e favorire l'utilizzo del servizio anche fuori da casa propria, ad esempio da un amico o dalla fidanzata.

Le informazioni di profilo sono visibili al ristorante ad eccezione della password di login che è protetta da un algoritmo crittografico SHA implementato sul server php, il quale cripta le password in fase di registrazione e le decripta in fase di login. Il database accessibile dal ristorante contiene quindi nel campo "password" una stringa incomprensibile che tutela la privacy di ogni utente.

2.3 La progettazione

Nell'attività di analisi tecnica e di progettazione si è verificata la necessità di definire le linee essenziali della struttura del software in funzione dei requisiti analizzati precedentemente. In questa fase è stata sviluppata la struttura di massima del progetto in base alle caratteristiche che l'applicazione doveva assumere, sono stati inoltre scomposti tutti i problemi principali in tanti moduli, per avere una chiara idea di come

e cosa implementare nella fase successiva. Per ottimizzare i tempi di sviluppo è stata data molta importanza e di conseguenza molto tempo a questa fase; essendo infatti la mole di lavoro molto grande, era necessario capire bene i limiti e i compromessi di ogni scelta organizzativa, per non perdere tempo a riscrivere del codice sbagliato o progettato in modo approssimativo. L'attività seguita nello svolgere la progettazione del software è riconducibile e divisibile approssimativamente in queste cinque fasi:

- Progettazione e analisi delle schermate di base
- Progettazione della struttura software.
- Scelta delle architetture software per un implementazione efficace
- Progettazione concettuale della base di dati
- Scelta della piattaforma e del linguaggio del server

A seguito verrà proposto un approfondimento delle fasi appena accennate.

2.3.1 Progettazione e analisi delle schermate di base

L'analisi e la progettazione delle schermate di base è un passo importante nell'inizio di un progetto, fortunatamente siamo aiutati in questa fase dal fatto che prima di essere sviluppatori siamo noi stessi degli utenti e sappiamo che chiunque preferisce un'interfaccia semplice ad una complessa a parità di funzionalità offerte, in questa fase quindi ho progettato il design delle schermate, impegnandomi a creare una struttura interattiva semplice ed efficace, che fosse intuitiva e non fraintendibile. Mi sono messo nei panni dell'utente medio, inesperto e non propenso a terminologie informatiche per capire quali dovessero essere le schermate principali dell'app e come gestire le funzionalità alle quali consentono l'accesso. Vista la natura del software l'obiettivo primario è stato quello di rendere utilizzabile l'app in qualsiasi momento, prevedendo ogni condizione di utilizzo, con o senza registrazione, con o senza connessione attiva. Si è inoltre deciso di mostrare alcuni dei servizi offerti prima di richiedere qualsiasi immissione di dati quali login e registrazione, poiché

nessun utente è molto propenso all'inserimento di dati personali, senza valutare prima il contenuto e i servizi offerti dall'applicazione. La progettazione delle schermate è iniziata dunque dando priorità alle schermate che non necessitano di registrazione, quali la Home, il modulo "Locale" ed il modulo "menu", per concludere con le altre.

2.3.2 Progettazione della struttura software

In questa fase della progettazione i problemi più importanti sono stati scomposti in moduli più piccoli, indipendenti tra loro. L'obiettivo principale è quello di ottenere una serie di documenti redatti sia formalmente secondo, ad esempio alle specifiche UML, sia in modo informale (pseudocodice); che possano fornire in modo chiaro una soluzione efficiente ad ogni necessità implementativa delle funzionalità dell'app. Un esempio importante sta nella produzione di diagrammi di flusso che rappresentano le principali interazioni dell'app. Un diagramma di flusso (FlowChart) si può definire come un linguaggio di modellazione grafico per rappresentare il flusso di controllo ed esecuzione di algoritmi, procedure, istruzioni operative; esso consente di descrivere in modo schematico le operazioni da compiere, rappresentate mediante sagome convenzionali e anche la sequenza nella quale devono essere compiute. Sono state inoltre progettate tutte le classi del progetto e tutte le activity, ovvero quelle classi che gestiscono l'interfaccia utente di un'applicazione il quale scopo è quello di permettere l'interazione con l'utente. La progettazione delle classi e delle activity per "moduli" è importante per l'ottimizzazione del progetto e rende più agevole eventuali porting e riscritture del codice su altre piattaforme e su altri sistemi operativi, infatti questo "step" di sviluppo non prende in considerazione il sistema operativo scelto per lo sviluppo dell'app.

2.3.3 Architetture software per un implementazione efficiente

Questo step di progettazione è di fatto molto vicino alla fase di implementazione del progetto, legato strettamente alla precedente fase di progettazione. In questa fase sono

state studiate le “possibilità realizzative” delle funzionalità richieste e valutate le varie alternative di implementazione a disposizione.

Si è quindi cercato di ottimizzare i tempi dedicati all’implementazione, fissando un’idea chiara e completa di tutte le strutture da utilizzare, un esempio che possa chiarire l’importanza di questo lavoro lo si può trovare nella scelta delle modalità di interrogazione del server da parte dell’applicazione per il download e l’invio dei dati e nella suddivisione dei compiti tra client e server nella formulazione delle query SQL; un ulteriore esempio consiste nella scelta dei pattern di programmazione da utilizzare nella stesura del codice.

2.3.3.1 Pattern

Nella fase progettuale appena descritta si fa riferimento alla scelta e allo studio dei pattern. Essendo questa fase molto importante nella progettazione del software è necessario spendere qualche parola in più sull’argomento.

In ingegneria del software, un design pattern è di fatto una soluzione progettuale generale ad un problema ricorrente. Si tratta di un modello logico da applicare per la risoluzione di un problema che può presentarsi in diverse situazioni durante le fasi di progettazione e sviluppo del software, necessariamente prima della definizione dell’algoritmo risolutivo della parte implementativa. In parole povere, l’obiettivo che ci si pone adottando un’analisi e progettazione accurata dei pattern è quello di standardizzare e incapsulare il codice il più possibile per lasciare all’implementazione il solo compito di eseguire la programmazione per moduli di ciò che è stato già progettato in toto in una fase precedente. Questa scelta progettuale ha il vantaggio di rendere i progetti object-oriented più flessibili e riutilizzabili e rendere il codice molto più comprensibile in caso di modifiche future o di collaborazione in un team di sviluppo.

2.3.4 Progettazione concettuale della base di dati

La parte sicuramente più importante nella fase progettuale è rappresentata dalla progettazione concettuale della base di dati. Il termine base di dati indica un insieme organizzato di dati. Le informazioni contenute in un database sono strutturate e collegate tra loro secondo un particolare modello logico scelto dal progettista del database tra i quali vanno citati ad esempio i modelli: relazionale, gerarchico, reticolare o a oggetti.

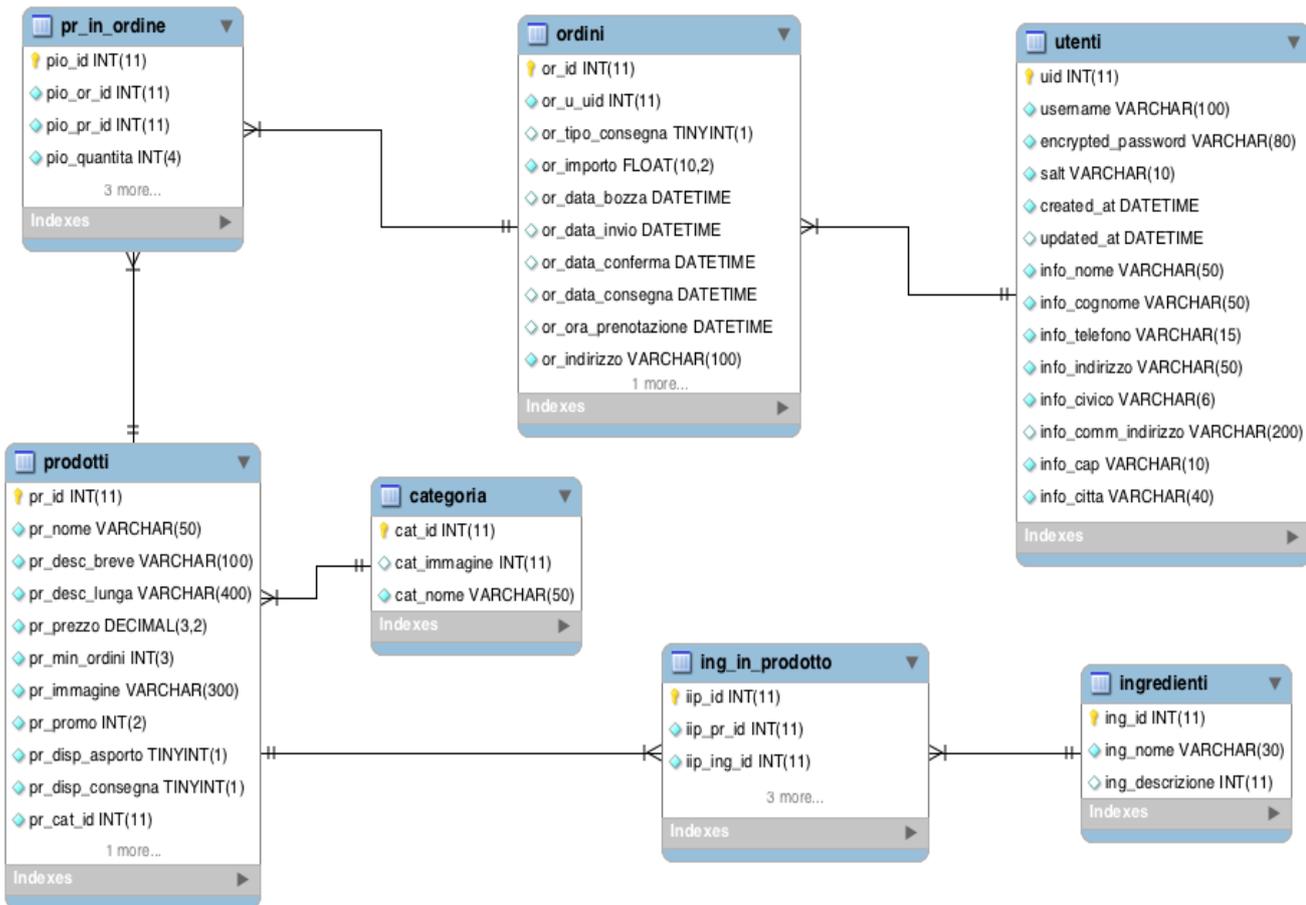
Gli utenti si interfacciano ai database attraverso i cosiddetti “query language” e grazie a particolari applicazioni software dedicate, ovvero i DBMS. Progettare concettualmente una base-dati significa individuare gli oggetti (o entità) che la costituiscono e le relazioni (o operazioni o associazioni) tra un oggetto e l'altro. Si tratta del livello più alto della progettazione di un data-base, quello più vicino all'uomo e più lontano dalla macchina (hardware). Deve quindi essere realizzata con strumenti e linguaggi comprensibili a tutti, non solo agli specialisti, e indipendenti dal sistema di data-base.

Il modello più utilizzato per la rappresentazione dell'architettura di un database è il modello E/R. Il modello entity-relationship o entità-relazione, è un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione. Viene utilizzato nella prima fase della progettazione di una base di dati in cui è necessario tradurre le informazioni risultanti dall'analisi di un determinato dominio in uno schema concettuale. Il modello E-R si basa su un insieme di concetti molto vicini alla realtà di interesse: quindi facilmente intuibili in fase di progettazione (e in genere considerati sufficientemente comprensibili e significativi anche per i non-tecnici), ma non implementabili sugli elaboratori. Infatti, pur essendo orientato alla progettazione di basi di dati, il modello prescinde dai criteri specifici di organizzazione fisica dei dati persistenti nei sistemi informatici.

I principali costrutti utilizzati nel modello E-R sono i seguenti:

- Entità: rappresentano classi di oggetti che hanno caratteristiche in comune. L'istanza di un'entità rappresenta un singolo oggetto appartenente a quell'entità.

- Associazione: rappresenta un legame tra due o più entità.
- Attributo: rappresenta una proprietà che descrive un'entità o un'associazione.
- Identificatore: costituisce un sottoinsieme di attributi di un'entità che identifica in maniera univoca ogni istanza della stessa entità, è indicato graficamente come un cancelletto prima del nome degli attributi che compongono l'identificatore.



[Figura 3 - Schema Entity/ Relationship]

Analizzando la struttura della base di dati progettata in questa fase possiamo identificare cinque entità principali (utenti, ordini, prodotti, categoria, ingredienti) e due entità di relazione per la reificazione di una relazione n ad n (pr_in_ordine,

ing_in_prodotto). A seguito verrà proposta una veloce analisi di tutte le entità progettate in relazione alle funzionalità che modellano.

L'entità utenti è un'entità di fondamentale importanza e, come si può facilmente intuire, si occupa di conservare i dati di tutti gli utenti registrati all'applicazione, gli attributi scelti nella progettazione di questa entità sono divisibili in 2 categorie, ovvero i dati di login, quali username, salt ed encrypted password e le informazioni utili all'applicazione per l'invio degli ordini. Gli attributi di login prevedono una crittografia della password eseguita sul server php secondo l'algoritmo crittografico SHA, mediante gli attributi password e salt. Gli attributi "created_at" e "updated_at" servono a memorizzare rispettivamente le date di creazione e modifica dei dati del profilo utente. La seconda categoria di attributi ai quali si è fatto riferimento precedentemente rappresenta informazioni utente necessarie per la domiciliazione e il contatto dell'utente quali nome, cognome, telefono, indirizzo completo, codice di avviamento postale e città. Questi dati sono considerati come informazioni di consegna "predefinite", ma sono in parte modificabili durante la creazione dell'ordine per agevolare gli ordini fatti fuori casa.

L'entità ordini rappresenta il fulcro dell'applicazione, si occupa di memorizzare qualsiasi ordine effettuato da ogni utente registrato oppure ospite. Gli attributi che caratterizzano questa entità sono:

or_id, chiave primaria identificativa dell'ordine, si tratta di un numero intero progressivo autoincrementale;

or_u_uid, chiave esterna di Join con la tabella utenti, serve a memorizzare l'utente che ha eseguito l'ordine;

or_tipo_consegna, è un parametro di tipo Booleano che può assumere i valori 0 e 1 per indicare se la tipologia di ordine scelta sia "consegna" o "asporto", questo parametro assume il valore 0 per default, ovvero esprime come tipologia di default la consegna;

or_importo, è un parametro booleano che memorizza l'importo totale dell'ordine, è stato implementato principalmente per evitare di sovraccaricare il database con inutili query di somma dei prodotti ordinati per i rispettivi prezzi;

or_data_bozza, or_data_invio, or_data_conferma, or_data_consegna, questi quattro attributi memorizzano le date di cambio stato dell'ordine, in base alla nullità di uno o più di questi attributi viene "calcolato" lo stato di avanzamento dell'ordine;

or_ora_prenotazione, è forse l'attributo più importante dell'entità, memorizza semplicemente l'ora di consegna della prenotazione;

or_indirizzo, è un attributo facoltativo, viene utilizzato in modalità "consegna" solo se l'utente desidera specificare un indirizzo di consegna diverso da quello registrato nel profilo utente.

L'entità prodotti ha un duplice utilizzo all'interno dell'applicazione, in ordine di utilizzo, serve inizialmente a mostrare il menu proposto dal ristorante, successivamente, all'interno del modulo "ordina" viene utilizzata a popolare la schermata delle pietanze ordinabili dall'utente. Gli attributi scelti per questa entità sono:

pr_id, chiave primaria identificativa del prodotto, si tratta di un numero intero progressivo autoincrementale.

pr_nome, esprime semplicemente il nome della pietanza;

pr_desc_breve, memorizza una breve descrizione del prodotto;

pr_desc_lunga, memorizza una descrizione più accurata della pietanza, questo campo viene principalmente utilizzato per descrivere informazioni quali, tecniche di cottura, ingredienti e valori nutrizionali;

pr_prezzo, memorizza il prezzo unitario della pietanza;

pr_min_ordini, questo attributo facoltativo, se specificato indica la quantità minima ordinabile per ogni pietanza, se prevista dal ristorante;

pr_immagine, link ad una risorsa memorizzata sul server, è una foto presentativa della pietanza scelta dal ristorante per presentare al meglio ogni prodotto;

pr_promo, questo parametro è progettato per prevedere la possibilità di creare promozioni che possano incentivare gli utenti all'acquisto di quantità maggiori ad un prezzo vantaggioso o mediante qualche omaggio. L'utilizzo di questo parametro è solamente una predisposizione ad una funzionalità che non è stata implementata nell'applicazione per motivi di semplicità di utilizzo;

pr_disp_asporto, indica la disponibilità della pietanza in modalità asporto;

pr_disp_consegna, indica la disponibilità della pietanza in modalità consegna;

pr_cat_id, chiave esterna che permette il join con la tabella categoria per la memorizzazione della categoria di appartenenza della pietanza in questione.

L'entità categoria, non la si può certamente classificare come una tra le entità più importanti, è stata progettata più che altro per ottimizzazione della quantità di informazioni da memorizzare sul database. Senza analizzare gli attributi che la compongono ci si può limitare a dire che è stata implementata per evitare di aggiungere ad ogni ordine l'attributo "categoria" che avrebbe creato importanti ridondanze. Ovviamente questa entità esprime la categoria del prodotto in questione. La divisione dei prodotti in categoria implementate nell'app TakeAway sono 3, ovvero "pizze, bibite e dessert", ma sono estendibili e modificabili in ogni momento.

L'entità Ingredienti, è stata scelta per memorizzare, per ogni pietanza pietanze offerta una descrizione accurata delle materie prime utilizzate, è composta da tre attributi che memorizzano un id, un nome e la descrizione della materia prima in questione.

Le ultime due entità da analizzare hanno una caratteristica comune, sono infatti delle entità associative e meritano una breve premessa.

Nella progettazione di database relazionali, le relazioni "molti a molti", ovvero relazioni dove una o più righe di una tabella sono associate a una o più righe di un'altra tabella, non sono consentite. Si consideri l'esempio della gestione delle fatture. Per aggirare il problema di avere una relazione "molti a molti" è necessario suddividerla in due relazioni "uno a molti". Questo è possibile utilizzando una terza tabella, una "tabella associativa". Ciascun record della "tabella associativa" contiene i campi chiave esterna delle due tabelle che unisce. Non è necessaria alcuna operazione particolare sui campi chiave esterna della tabella associativa, poiché vengono popolati con i dati delle altre due tabelle mano a mano che i record vengono creati. Non è insolito che una tabella associativa contenga numerosi record, poiché questi vengono creati nella tabella associativa mano a mano che vengono creati dei record nelle due tabelle che unisce.

pr_in_ordine, questa tabella si occupa di relazionare le tabelle ordini e prodotti per potere memorizzare i prodotti ordinati e la relativa quantità per ogni ordine;

ing_in_prodotto, similmente alla tabella appena descritta, questa tabella mette in relazione i prodotti agli ingredienti che li compongono.

2.3.5 Scelta della piattaforma e del linguaggio del server

La progettazione del web-server al quale si interfaccia l'applicazione è una fase molto importante nello sviluppo dell'app. In questa fase vengono principalmente scelti gli strumenti sui quale basare i servizi necessari richiesti dall'applicazione. Svolgendo questo lavoro come ultima fase di progettazione, dopo cioè aver progettato l'architettura del software e dopo aver definito la base di dati è chiaro come l'obiettivo da raggiungere sia rendere possibile e successivamente ottimizzata la comunicazione tra l'applicazione ed il server. In questa fase finale del processo progettuale sono stati ripartiti i compiti tra applicazione e server nella formulazione delle richieste e nella manipolazione delle risposte. Un'analisi più approfondita del "lato Server" verrà argomentata a breve durante la presentazione della fase implementativa del progetto.

Capitolo 2 - L'implementazione

L'implementazione del progetto è la fase finale dello sviluppo di un'applicazione, in questa fase viene realizzato il codice sorgente che concretizza tutte le funzionalità richieste dall'applicazione, seguendo in modo rigoroso la documentazione prodotta in fase progettuale. Le prime scelte in questa fase sono le più importanti da compiere e condizionano profondamente il progetto durante tutta la fase implementativa. In questo capitolo verrà descritta l'implementazione del progetto TakeAway su piattaforma Android.

3.1 La scelta del sistema operativo

La prima scelta da compiere prima di iniziare questa fase del progetto riguarda la piattaforma che ospiterà l'applicazione. Questa scelta è molto importante perché condizionerà il lavoro fino alla fine dello sviluppo del progetto. Inoltre ogni sistema operativo mobile ha diverse peculiarità e caratteristiche e adotta scelte architetturali e politiche di sviluppo diverse. Prima di analizzare questa prima fase dell'implementazione del software occorre fare una premessa per conoscere ed analizzare i tre sistemi operativi mobile di riferimento.

3.1.1 Android

Android nasce 10 anni fa, quando Andy Rubin, Rich Miner, Nick Sears e Chris White fondano Android Inc. Il progetto Android nasce come sistema operativo per fotocamere digitali, ben presto però, i quattro decisero di cambiare rotta e virare verso il mercato degli smartphone, convinti che gli smartphone avrebbero sopraffatto le fotocamere. L'avventura di Android, sembra però avere vita breve. Nel corso del primo anno i fondi stanziati dai fondatori finirono e il progetto rischiò l'archiviazione. In questa occasione fu di fondamentale importanza l'intervento di Steve Perlman,

amico di Rubin ed ex ingegnere Apple, il quale diede nuove speranze alla piccola società con una donazione di 10000 dollari. Nel 2005 fa la sua apparizione Google che decide di finanziare il progetto e dopo pochi mesi, ne prende il controllo. Dopo oltre un anno di anonimato, Android torna a far parlare di sé nel 2007. Nello stesso anno della presentazione del primo iPhone, Google annuncia la costituzione dell'Open Handset Alliance (OHA), un consorzio che riuniva una lunga lista di produttori di smartphone e telefonini, con lo scopo di gettare le basi per lo sviluppo di standard aperti in ambito mobile. È l'occasione di presentare al mondo il suo sistema operativo mobile: Android. Nel febbraio 2008 Qualcomm e Texas Instruments annunciano la produzione dei primi chip compatibili con l'architettura software di Android. Il 28 ottobre di questo stesso anno HTC annuncia il lancio del modello **Dream** nato dalla collaborazione del produttore sudcoreano con Google e T-Mobile. Si tratta del primo smartphone Android della storia. Il 30 aprile 2009 Google rilascia Android 1.5 Cupcake e nel secondo trimestre di questo stesso anno Android si attesta al 2,9% del mercato internazionale e inizia a fare proprie quote di mercato a iOS, Symbian e tutti gli altri. Dopo altre tre release distribuite nell'arco di pochi mesi a dicembre Android è installato sul 33% dei cellulari presenti sul mercato mondiale, mentre negli Stati Uniti ha già sorpassato iOS. Nel secondo trimestre dell'anno successivo Android passa la soglia "psicologica" del 50% del mercato, divenendo il sistema operativo mobile più diffuso al mondo.

3.1.2 iOS

Apple iOS è un sistema operativo per dispositivi mobili progettato da Apple e utilizzato nella gamma iPhone, iPod e iPad. Il sistema operativo è stato presentato il 9 gennaio 2007 al Macworld Conference & Expo di San Francisco, e la versione 1.0, ancora priva di nome, è entrata in commercio con il primo iPhone nello stesso anno. Il 6 marzo 2008, in concomitanza con la pubblicazione della prima beta del SDK, il sistema operativo è stato denominato ufficialmente come "iPhone OS".

Nel luglio 2008 viene pubblicato in concomitanza della vendita di iPhone 3G l'aggiornamento a iPhone OS 2.0 che aggiunge, tra le altre funzioni, il molto atteso App Store e la possibilità di installare applicazioni di terze parti tramite l'app.

iPhone OS 3.0, pubblicato con l'iPhone 3GS il 17 giugno 2009 ha aggiunto molte funzioni che furono richieste dagli utenti, alcuni dei quali il copia e incolla e gli MMS. Tutti i dispositivi erano aggiornabili a iPhone OS 3, ma con delle limitazioni per la prima generazione di iPhone e iPod touch. Il primo iPad, entrato in commercio nell'aprile 2010, ha avuto inizialmente un "ramo" separato di iPhone OS 3, fino all'unificazione con gli altri dispositivi con la versione 4.2.1 del software.

La quarta versione del sistema operativo, rilasciata con iPhone 4 nel 2010, ha aggiunto numerose funzioni quali il multitasking per le applicazioni di terze parti, FaceTime e iBooks. La versione corrente di iOS è la 8. Allo stesso modo del sistema operativo Mac, anche iOS è una derivazione di UNIX e usa un microkernel XNU Mach basato su Darwin OS a quattro livelli di astrazione. Come anticipato il negozio virtuale da cui è possibile scaricare app destinate ai dispositivi Apple è l'App Store, il quale vede la luce nel 2008 e permette lo scaricamento di applicazioni disponibili all'interno dell'iTunes Store. I software possono essere gratuiti o a pagamento e riservati ai proprietari di dispositivi mobili Apple quali iPhone, iPod touch e iPad. Una stima risalente a giugno 2014 rileva che i download dallo Store risultano essere 75 miliardi mentre il numero di app disponibili si avvicina a 1.200.000.

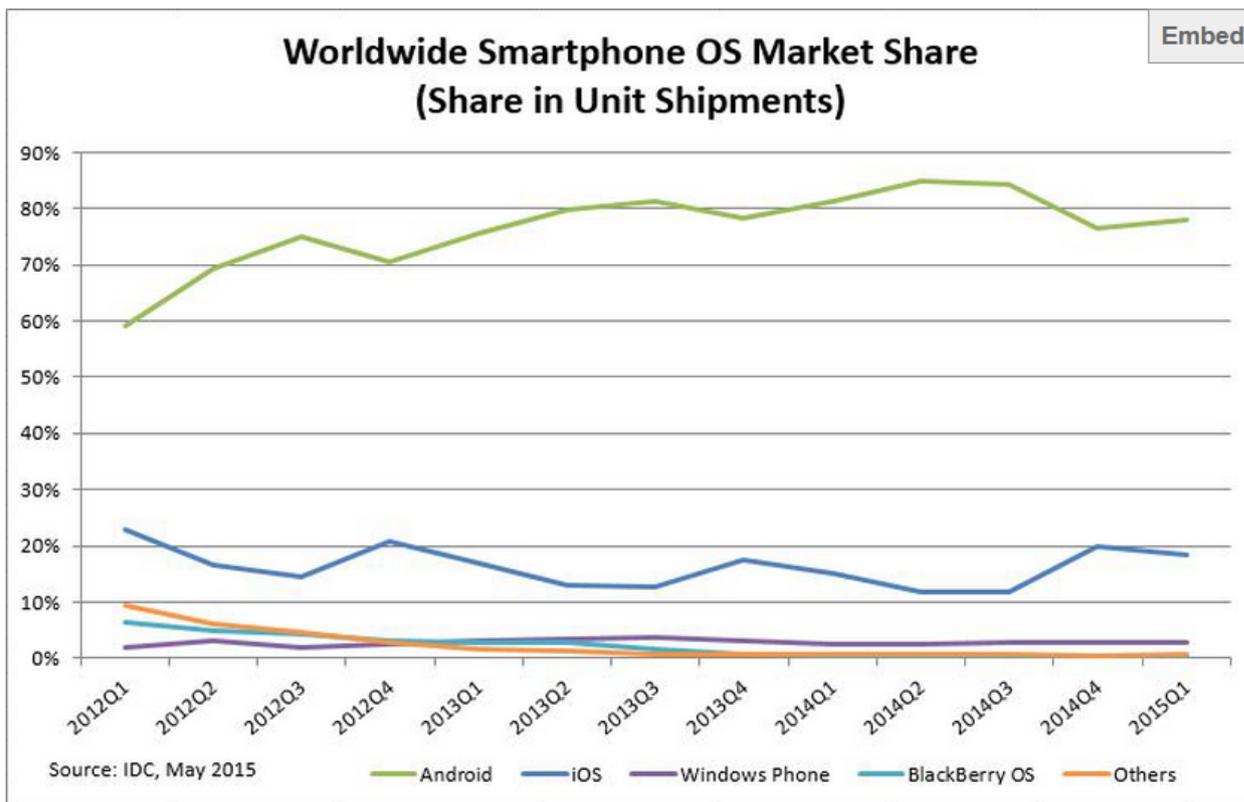
3.1.3 Windows Phone

Windows Phone è il sistema operativo sviluppato da Microsoft e presentato nel 2010, successore di Windows mobile, con il quale non garantisce nessuna retrocompatibilità, nonostante ci siano molte analogie con il predecessore. Dopo anni di dominio nel mondo mobile di fascia alta con Windows CE (Mobile), in cui ha sfiorato il 50% del mercato, Microsoft si è trovata del tutto impreparata a fronteggiare

le nuove sfide dettate dalla conquista del mercato consumer degli Smartphone e dei due principali sistemi operativi che li hanno accompagnati: Apple iOS e Google Android.

La casa di Redmond si è vista ridurre ad una quota insignificante la propria presenza ed ha dovuto innovarsi drasticamente per tentare di recuperare il terreno perduto. Così nasce Windows Phone 7 (WP7) che sostituisce Windows Mobile 6.x ed introduce una nuova UI basata su Tiles e design essenziale aprendo così le porte ad un modo completamente diverso di concepire l'utilizzo degli smartphone rispetto ai diretti concorrenti. WP7 è però ancora basato sul Kernel CE ed è solo con Windows Phone 8 / 8.1 che tale legame si spezza definitivamente, essendo la nuova versione realizzata sulla stessa base di Windows 8, ovvero il cosiddetto Kernel NT. L'ultima versione distribuita è la 8.1, con la quale Microsoft cerca di unificare lo sviluppo di app per PC e lo sviluppo di app per piattaforma mobile. Il negozio virtuale dal quale scaricare software per Windows Phone è il Windows Phone Store il quale, a fine 2013, contava “appena” 190.000 app.

Il sistema operativo mobile scelto per lo sviluppo dell'applicazione TakeAway è Android. Questa scelta è motivata dalle caratteristiche opensource del progetto Android e dal market-share che fa di Android il primo sistema operativo mobile a livello mondiale; questo porta a numerosi vantaggi, quali ad esempio: una grande quantità di contenuti online ideali per un apprendimento continuo delle tecniche di programmazione utilizzate e non per ultimo, la migliore gestione della documentazione per programmatori fornita da google per lo sviluppo di applicazioni android.



[Figura 4 – MarketShare OS Mobile secondo IDC, Maggio 2015]

3.2 Analisi dell'implementazione (lato Client)

Organizzazione

Durante la fase implementativa si è scelto di dividere il carico di lavoro in due attività afferenti all'implementazione: l'implementazione dei singoli moduli che costituiscono il sistema completo e la successiva integrazione di tali moduli a formare il sistema complessivo.

Tecnologia

La fase di implementazione coinvolge spesso numerose tecnologie relative non solo al prodotto, ma anche al processo che lo realizza per ottimizzare e rendere più agevole per il programmatore la fase di sviluppo del codice sorgente.

Per quanto riguarda il prodotto, la stesura del codice necessita di un linguaggio di programmazione che nello sviluppo di applicazioni Android si identifica in JAVA.

JAVA è un linguaggio di programmazione orientato agli oggetti, nato nei primi anni novanta e divenuto primo linguaggio al mondo per diffusione. Durante la fase implementativa verranno utilizzati altri linguaggi di programmazione quali php per il web-server e SQL per la base di dati.

Il processo di implementazione non può inoltre prescindere da un editor per la scrittura del codice sorgente e di un compilatore o un interprete per collaudare l'esecuzione del codice. La scrittura della documentazione sul prodotto, invece, può essere generata in maniera automatica a partire dai commenti scritti all'interno del codice sorgente attraverso tool opportuni. Spesso la maggior parte delle tecnologie necessarie alla produzione sono disponibili in un ambiente di sviluppo integrato, un'applicazione che unifica gli strumenti necessari al programmatore, o in kit di sviluppo (SDK), una distribuzione di documentazione e implementazione di un linguaggio di programmazione o piattaforma.

L'ambiente di sviluppo (IDE) scelto per la programmazione è Eclipse, l'alternativa più diffusa ad Android Studio, IDE ufficiale creato da Google, ma ritenuto da tanti ancora molto instabile e poco parsimonioso di risorse hardware. Queste due considerazioni giustificano la scelta di utilizzare il primo piuttosto che il secondo. Il test dell'applicazione è avvenuto sempre su dispositivo fisico, nonostante Eclipse offrisse macchine virtuali per il testing su PC.

L'implementazione del progetto TakeAway sarà analizzata illustrando le

caratteristiche delle varie Activity (schermate) di cui si compone l'app spiegandone sinteticamente il funzionamento.

3.2.1 SplashActivity

La prima schermata che si può osservare all'apertura dell'app è la SplashActivity, ovvero una Activity che mostrando il logo dell'app su sfondo rosso, esegue in background il download o il caricamento locale dei contenuti da mostrare nell'app. E' stata progettata per gestire tutti i casi in cui si può incorrere all'apertura dell'app; nel dettaglio, viene verificata la disponibilità di connessione ad internet, wifi o rete mobile per decidere se eseguire il download dal server dei contenuti più recenti oppure caricare i contenuti già scaricati all'avvio precedente. Se il primo avvio dell'app avviene in assenza di connessione internet, è previsto il caricamento di contenuti di default pre-caricati nell'applicazione o scaricati in una sessione precedente. Eseguite tutte le operazioni di controllo della connessione vengono inizializzate e preparate tutte le risorse necessarie all'applicazione come immagini e dati testuali. Terminata questa fase di preparazione, quando l'app è pronta l'utente si troverà nella schermata principale dell'applicazione, la HomeActivity. Se l'applicazione sarà lanciata senza una connessione internet attiva, essa potrà essere utilizzata in modalità offline, non saranno disponibili il login, la registrazione e la possibilità di eseguire ordini, si potranno però utilizzare tutte le funzioni di presentazione del locale e del menu.



[Figura 5 – SplashActivity]

3.2.2 HomeActivity

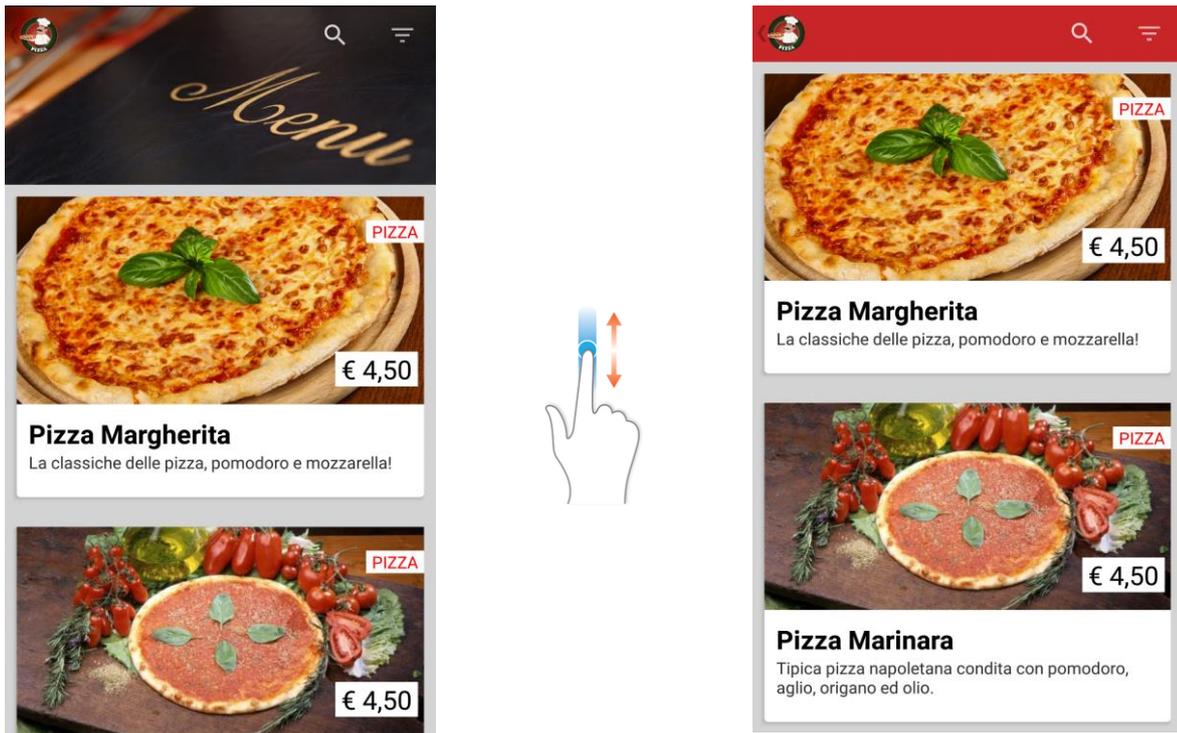
E' la schermata principale dell'app, da qui l'utente può raggiungere tutte le altre schermate e navigare tra tutti i contenuti disponibili. Questa activity si presenta con un'immagine a tutto schermo rappresentante la categoria merceologica dei prodotti offerti (nell'esempio presentato è una pizza) sopra la quale è presente, nella parte alta, il logo e nome del ristorante, mentre nella parte bassa i quattro link ai moduli Ordine, Menu, Locale, Utente. E' stata progettata in modo che risultasse graficamente accattivante, ma semplice da utilizzare, per questo sono stati utilizzati font chiari e caratteri grandi, con icone ben visibili e intuitive.



[Figura 6 – Home]

3.2.3 MenuActivity

Questa Activity mostra all'utente il menù del locale, è implementato attraverso una ScrollView per ottimizzare le prestazioni anche su device un po' datati. In questa schermata a scorrimento verticale possiamo visualizzare tutte le pietanze offerte con una grande foto dimostrativa, una breve descrizione del piatto e una seconda descrizione approfondita degli ingredienti e delle tecniche di cottura. Le pietanze sono elencate con uno stile a carte dove ogni "carta" oltre ai dati prima descritti, presenta un indicatore in sovraimpressione con il prezzo e la categoria del prodotto. Nella parte alta della schermata è presente un'immagine di presentazione dotata di un effetto grafico a scomparsa che ad ogni scorrimento diventa una barra operativa con funzionalità di filtro categoria e di ricerca tra i prodotti.



[Figura 7 – Menù]

3.2.4 LocaleActivity

LocaleActivity è la schermata che presenta all'utente il locale in toto. In questo modulo sono presenti tutte le informazioni sul ristorante, a partire dai contatti alle foto di interni ed esterni, passando per localizzazione del ristorante, fascia di prezzo e servizi offerti. Nella parte alta della schermata si può trovare un'immagine con il nome del ristorante sotto al quale sono inserite delle icone interattive per effettuare con un tap chiamata diretta, sms, mail o visualizzazione del sito internet (se presente).



DESCRIZIONE

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

FOTO



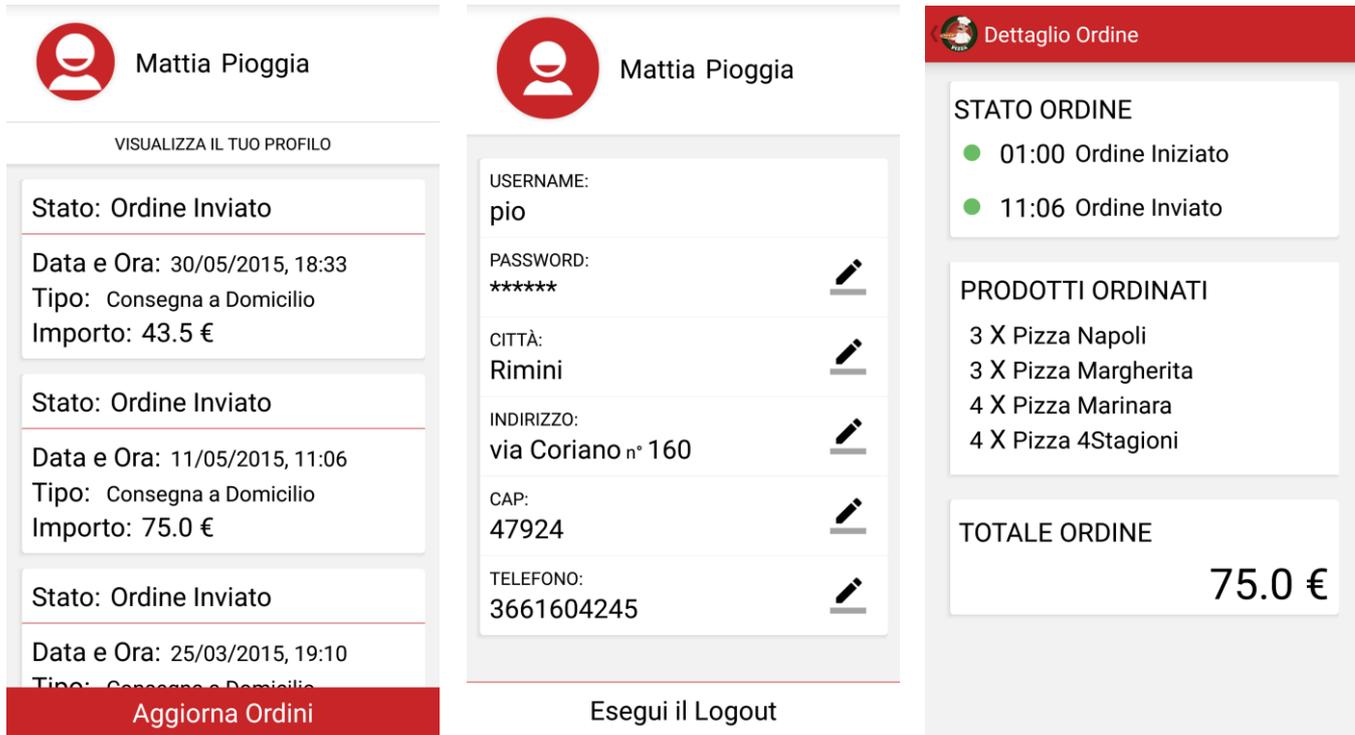
INFORMAZIONI

● Conseqna a Domicilio

[Figura 8 - Locale]

3.2.4 UtenteActivity

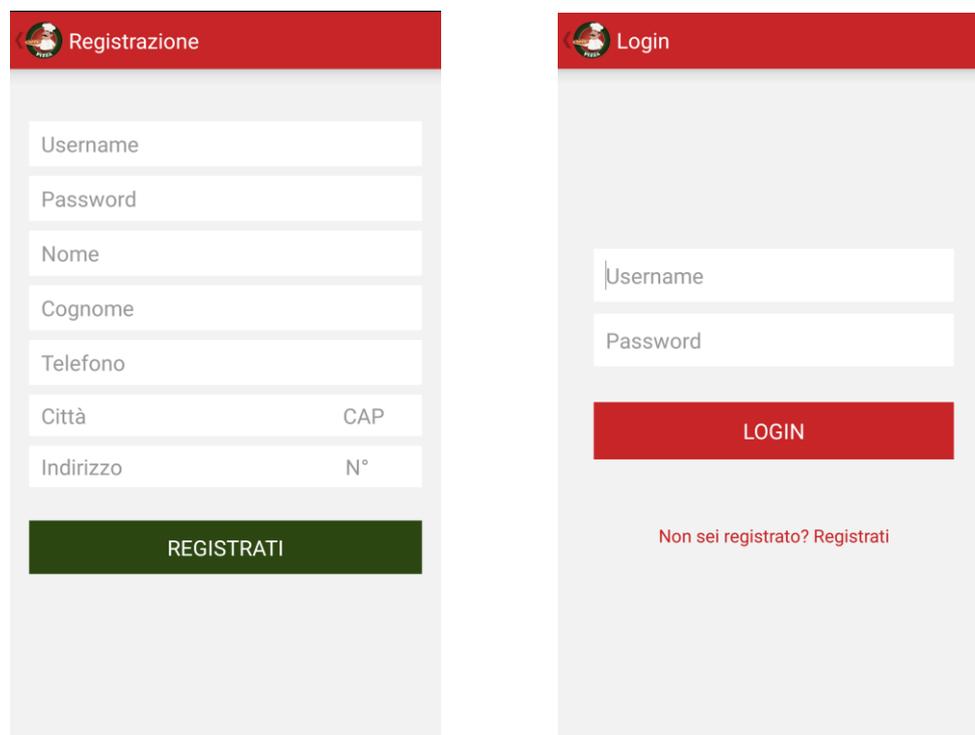
Questa schermata consente all'utente di visualizzare tutti gli ordini effettuati, con le informazioni principali attraverso una scrollView semplice, con l'azione di "tap" sul singolo ordine verrà visualizzata una pagina con tutte le informazioni disponibili sul dato ordine. La parte alta della schermata è caratterizzata da una barra orizzontale, la quale mostra il nome utente, il "tap" su di essa da accesso a tutti i dati utente, permettendo la visualizzazione e la modifica degli stessi. La parte bassa dell'activity contiene una sezione "Aggiorna" che permette di sincronizzare la lista ordini con il server.



[Figura 9 - Utente]

3.2.5 Login e Registrazione

Queste due Schermate gestiscono il login e la registrazione dell'utente. Sono pagine molto semplici dal punto di vista grafico, ma non banali dal punto di vista implementativo. In entrambi casi i dati vengono inseriti in "EditText" che passano i dati all'activity per generare una richiesta http di tipo post. Per il login la richiesta ricevuta dal server genera un'interrogazione al DB per il controllo di username e password, se l'esito è positivo l'utente verrà indirizzato al proprio profilo dove può visualizzare gli ordini effettuati o modificare i dati utente. Nel caso della registrazione, quando i dati utente raggiungeranno il server, dopo un controllo di non duplicità, l'utente sarà aggiunto al database e l'applicazione Android eseguirà il login automatico con i dati appena inseriti.

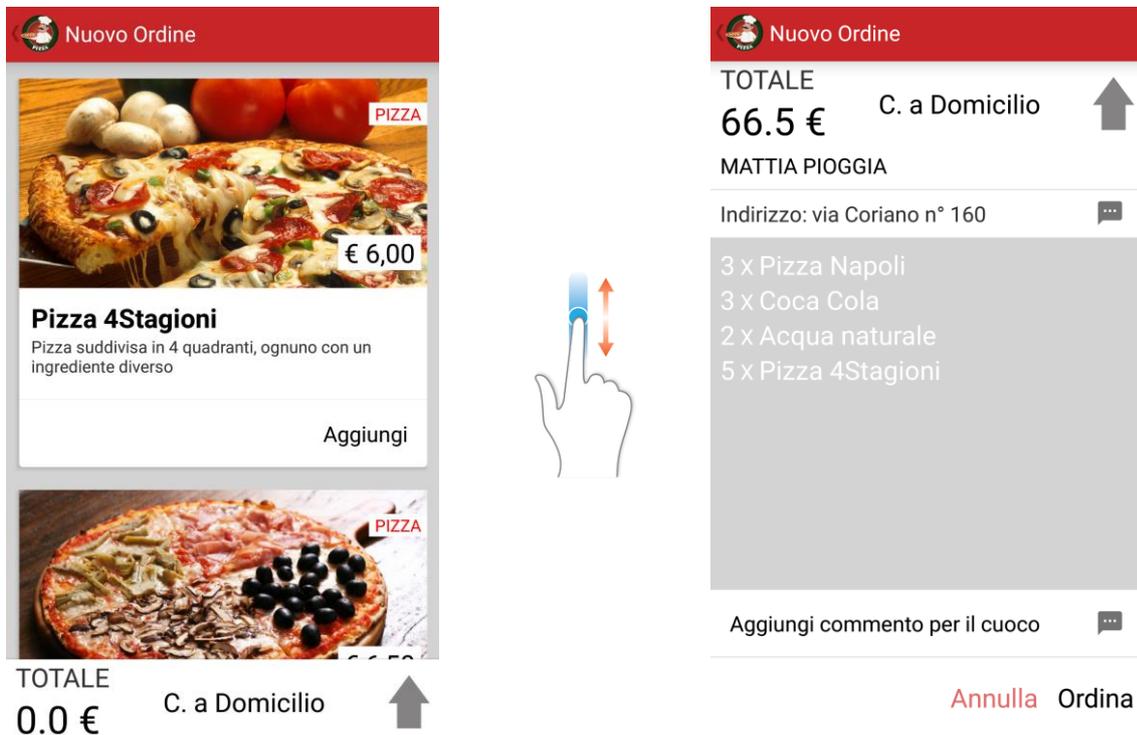


[Figura 10 – Login e Registrazione]

3.2.6 OrdineActivity

E' sicuramente l'activity più importante e più complessa dell'app. Si occupa di visualizzare un menu a scorrimento simile a quello della sezione "Menu", con la differenza che da questa schermata possiamo creare il nostro ordine aggiungendo e rimuovendo le pietanze. Dopo la selezione dei prodotti da ordinare con la relativa quantità, un pannello scorrevole verticalmente mostra il totale, il riepilogo dell'ordine e i dati dell'utente. L'utente prima della conferma può modificare alcuni dati personali per quell'ordine, ad esempio l'indirizzo; può inoltre aggiungere un commento per il cuoco, nel caso volesse comunicare qualche indicazione ad esempio sulla cottura o su allergie a prodotti. Premendo sul bottone di conferma è possibile scegliere l'orario di consegna e successivamente confermare l'ordine. Inizialmente OrdineActivity segue un controllo di login, che se non presente viene richiesto, attraverso registrazione, login o accesso ospite. Una volta inviato, l'ordine viene inviato al server attraverso una chiamata http con metodo post. La query di inserimento nel database viene elaborata dal server php che, se non incontra errori,

notifica l'avvenuto inserimento. Un messaggio di conferma dell'ordine verrà visualizzato al cliente. L'ordine appena creato sarà visionabile e controllabile nel pannello utente insieme a tutti gli ordini effettuati precedentemente dall'utente.



[Figura 11 - Ordine]

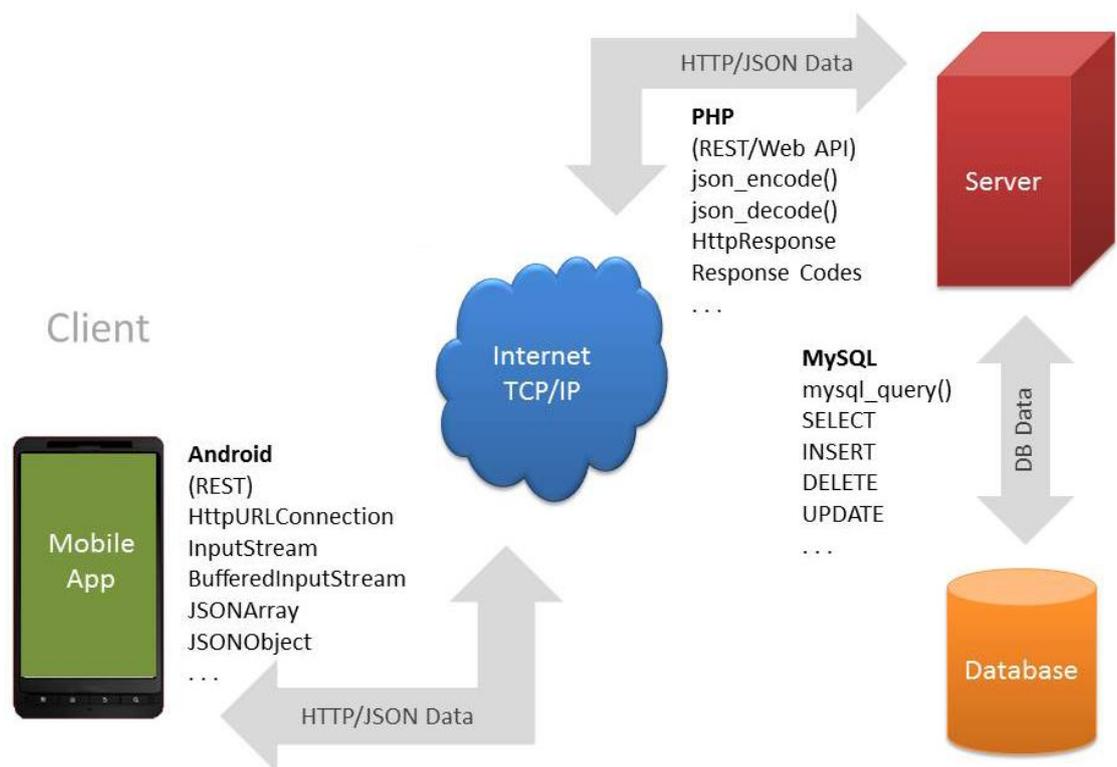
3.3 Analisi dell'implementazione (lato Server)

L'applicazione TakeAway offre il meglio delle proprie funzionalità sfruttando il web; è connessa ad internet ed ogni risorsa che sfrutta viene scaricata all'avvio aggiornando i dati da internet, questo è essenziale per garantire la correttezza dei dati mostrati all'utente in ogni momento (se viene eseguita in modalità online).

Tutti i dati necessari all'applicazione sono localizzati su un server online, accessibile in ogni momento. Il server contiene tutte le informazioni del locale, tutte le immagini

e tutte le risorse essenziali per il funzionamento dell'applicazione. Il server inoltre ospita la base di dati su cui verte l'applicazione; questa è composta dai dati utente, dalle pietanze, dagli ordini effettuati e da tutte le informazioni testuali utilizzate.

La terza risorsa che caratterizza il server sono le pagine php che gestiscono tutte le richieste, dopo la richiesta, tramite funzioni specifiche esse elaborano risposte attraverso query SQL.



[figura 12- Struttura Client/Server]

3.3.1 Web Hosting

Nella fase implementativa dei servizi web il passo principale è la scelta di un servizio di "hosting" che possa ospitare i dati e i servizi su un server sempre accessibile. Nella ricerca di un hoster che potesse soddisfare le necessità dell'applicazione, il miglior compromesso tra costi e servizi offerti è stato identificato in Altervista.

Altervista, piattaforma web, offre gratuitamente:

- un dominio personale;
- uno spazio limitato su cui caricare dati di ogni tipo;
- la possibilità di utilizzare pagine php;
- un database MySQL;
- Accesso FTP;

Il rovescio della medaglia sono le prestazioni limitate, ma comunque sufficienti a svolgere tutte le necessità dell'applicazione. L'iscrizione e la configurazione è molto semplice, per il caricamento dei dati e delle pagine su server è stato utilizzato Filezilla Client, un progetto opensource che permette il trasferimento di file inRete attraverso il protocollo FTP. Il programma è disponibile per GNU/Linux, Microsoft Windows, e Mac OS X. Tra i vari protocolli supportati, oltre all'FTP vi è l'SFTP, e l'FTP suSSL/TLS.

3.3.2 I servizi

Come anticipato, la gestione dei servizi viene garantita da pagine php, pronte a ricevere richieste HTTP di tipo post, specifiche per ogni informazione ricercata. Le pagine utilizzate sono cinque:

- Config.php : una pagina dove vengono semplicemente definiti i dati di accesso al database utilizzato, è utile scorporare questi dati dalla pagina che si occuperà della connessione al database per poter velocizzare le eventuali modifiche ai dati di accesso;

- DB_Connect: questa pagina contiene i metodi necessari alla connessione al database:
- Index.php : Questa è la pagina principale, si occupa di intercettare le richieste http, identificandole per mezzo di un tag nel quale viene specificato il tipo di richiesta es: registrazione utente, login, richiesta dati utente, appena viene ricevuta una richiesta se ne leggerà il contenuto. Le richieste sono in formato JSON, quindi una funzione si occuperà di decodificare il JSON suddividendo i parametri in array di dati contenenti tutte le informazioni necessarie. Tra questi dati ci sarà anche il “TAG”, che identifica la tipologia della richiesta. In base al tag richiesto, uno “SWITCH CASE” chiamerà il giusto metodo in grado di risolvere la richiesta, ottenuta la risposta, essa dovrà essere codificata in formato JSON prima di essere inviata al mittente con un “echo”;
- DB_Function: sono stati implementati tutti i metodi che si occupano di elaborare la richiesta e generare l’output, la risposta viene composta dopo un’interrogazione al database per mezzo di una query SQL

3.3.3 Standard per la comunicazione Client/Server

La comunicazione tra Client e Server, gestita dalle pagine php e dal codice JAVA necessita di uno strumento comune ai due linguaggi che sia “comprensibile” per entrambi con il fine di trasferire informazioni da un nodo all’altro in modo pratico, veloce e funzionale. Lo standard adottato per questo scopo è il formato JSON. JSON (JavaScript Object Notation) è un formato per lo scambio di dati. La sua enorme diffusione è dovuta principalmente alla facilità di implementazione per gli sviluppatori e analizzarne la sintassi. Si basa su un sottoinsieme del Linguaggio di Programmazione JavaScript.

JSON è un formato di testo completamente indipendente dal linguaggio di programmazione, ma utilizza convenzioni conosciute dai programmatori di linguaggi della famiglia del C, come C, C++, C#, Java, JavaScript, Perl, Python, e molti altri. Questa caratteristica fa di JSON un linguaggio ideale per lo scambio di dati.

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. E' sensato che un formato di dati che è interscambiabile con linguaggi di programmazione debba essere basato su queste strutture.

In JSON, assumono queste forme:

Un oggetto è una serie non ordinata di nomi/valori. Un oggetto inizia con “ { ” e finisce con “ } ” . Ogni nome è seguito da “ : ” e la coppia di nome/valore sono separata da una virgola.

Un esempio di file di configurazione iniziale in formato JSON. In questo esempio, il file non viene utilizzato per richieste client/server, ma per un veloce file di configurazione iniziale utilizzato quando l'app viene eseguita Offline, questo file corrisponde all'ultima richiesta http eseguita quando l'app era online e salvata su file a titolo di Backup dei dati di configurazione.

```

1  {
2    "menu": {
3      "pietanze": [
4        {
5          "id": 1,
6          "nome": "Pizza Margherita",
7          "descrizioneBreve": "La classiche delle pizza, pomodoro e mozzarella!",
8          "descrizione": "Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod ten
9          incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
10         ris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cill
11         olore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui offi
12         deserunt mollit anim id est laborum.",
13         "ingredienti": [
14           "Pomodoro",
15           "Mozzarella"
16         ],
17         "prezzo": 4.5,
18         "minOrdini": 1,
19         "immagine": "http://ftp.mattiapioggia.altervista.org/immagini/prodotti/pizze/margherit
20         g",
21         "promozioneAssocita": 0,
22         "disponibileAsporto": true,
23         "disponibileConsegna": true
24       },
25       {
26         "id": 2,
27         "nome": "Pizza Marinara",
28         "descrizioneBreve": "Tipica pizza napoletana condita con pomodoro, aglio, origano ed c
29         ",
30         "descrizione": "Lorem ipsum dolor sit amet, consectetur adipisci elit, sed eiusmod ten
31         incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
32         ris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cill
33         olore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui offi
34         deserunt mollit anim id est laborum.",
35         "ingredienti": [
36           "Pomodoro",
37           "Aglio",
38           "Origano",
39           "Olio"
40         ],
41         "prezzo": 4.5,
42         "minOrdini": 1

```

[Figura 13 – file JSON]

3.4 Librerie Utilizzate

Nell’implementazione del progetto è risultato necessario l’utilizzo di alcune librerie che agevolassero l’implementazione di alcune parti di codice. Le librerie utilizzate sono tutte di licenza opensource, fornite dalla community “github” o da google stessa.

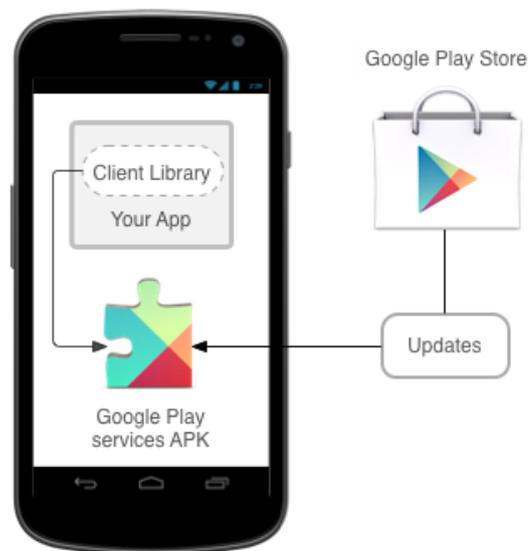
3.4.1 Google Play Services (client)

I Google Play Services sono un progetto software che offre in un unico pacchetto librerie, API e funzionalità varie con cui le applicazioni presenti su un dispositivo Android possono interagire con i servizi web di Google in maniera ottimizzata, al

passo con gli ultimi aggiornamenti rilasciati ed in una modalità di sviluppo molto produttiva dal punto di vista del programmatore.

Per questo ogni dispositivo Android vede installati e costantemente aggiornati i Google Play Services al proprio interno.

Affinchè un nostro progetto possa sfruttarli per interagire con un servizio remoto di Google, deve includere un collegamento ad una particolare libreria "client" fornitrice di tutte le interfacce di comunicazione. Nel caso specifico, si è resa necessaria l'inclusione di questa libreria client all'interno del progetto TakeAway per sfruttare le funzionalità di localizzazione di Google Maps all'interno della sezione Locale, per la visualizzazione della mappa, indicando la posizione del ristorante.



[Figura 14 – Schema Google Play Services]

3.4.2 Librerie grafiche

L'applicazione fa un importante uso di effetti grafici di scorrimento nelle sezioni "Locale", "Menu" e "Ordine" gli effetti di transizione basati sullo scrolling sono stati implementati sfruttando alcune librerie sviluppate in licenza opensource volte a

emulare gli effetti di transizione adottati di recente dai servizi Google, nello specifico le due librerie utilizzate sono: `fadingActionBar` utilizzata nel menu e `slidingUpPanel` utilizzata per il riepilogo ordine nella sezione “Ordina”.

Capitolo 3 – Conclusione

La realtà italiana in ambito di applicazioni legate alla ristorazione e al Take Away è ancora acerba, in poche città hanno preso piede queste applicazioni/siti web, forse per un'offerta troppo frammentata o forse per una pigrizia tecnologica che richiede un'azione sociale forte e decisa. Si potrebbe pensare che questo genere di applicazioni possa essere poco utile all'utente medio, ma esempi di successo nel mondo ci dicono il contrario, nel resto del mondo, soprattutto in USA e UK troviamo casi di applicazioni simili al progetto TakeAway che hanno avuto un successo esplosivo negli ultimi 2-3 anni. Ciò indica che la rivoluzione del settore arriverà sicuramente anche in Italia e questo è il momento giusto per investire su questo genere di progetti. Il progetto TakeAway fu oggetto di tirocinio formativo, sfociato poi in un progetto più grande ed ambizioso al centro di questa tesi. Come primo step è stata creata l'applicazione android priva di database, dopodiché è stato implementato il database e i servizi online, una gestione completa degli ordini e per finire la possibilità di registrarsi e modificare i propri dati. Il progetto ha dimostrato la fattibilità dello sviluppo di un'app semplice e adatta a tutti che unisse tutte le tecnologie disponibili nel mondo mobile e web, per uno scopo innovativo e precursore dei trend che porranno al centro della vita quotidiana la comunicazione tramite app e siti web. L'esperienza ha fornito la possibilità di approfondire ed esplorare molti ambiti della programmazione mai affrontati in precedenza in prima persona. Prime tra tutte, l'integrazione e la conciliazione di tecnologie diverse e strumenti diversi per il raggiungimento di importanti traguardi nello sviluppo di applicazioni mobile, che risultano complete sotto ogni punto di vista e appetibili per l'utente.

3.1 Sviluppi futuri

In questo momento l'applicazione è completa e utilizzabile su piattaforma Android, ma questo non è certamente sufficiente per avere un prodotto finito e appetibile. Gli sviluppi futuri riguarderanno certamente il porting dell'applicazione su piattaforma

iOs e Windows Phone, sarà poi interessante capire se potrebbe essere utile anche una versione web per i sistemi desktop. I principali progetti di riferimento esteri consigliano di non implementare una versione web, rimanendo nel mondo mobile, ma il progetto è molto allettante.

Riguardo invece, il miglioramento dei servizi offerti, sarà utile inserire la possibilità di eseguire il pagamento delle merci direttamente dal device, utilizzando il circuito Paypal. Sarà inoltre interessante integrare nell'applicazione un'utilizzo maggiore dei principali social network, per un'esperienza utente migliore. Un impegno futuro riguarderà sicuramente due ulteriori aspetti da migliorare, l'ottimizzazione delle prestazioni e la minimizzazione dei consumi di RAM. Inoltre esiste un importante margine di miglioramento nella gestione delle connessioni per limitare le latenze di caricamento delle risorse web, che già in questo stato sono buone. Infine è prevista una seconda applicazione/sito web che funzioni come back office per il ristoratore, ovvero un portale dal quale controllare gli ordini e modificare pietanze e informazioni con un user interface facilitata e alla portata di tutti.

3.2 Sitografia

- <http://www.idc.com/>
- <http://www.forbes.com/>
- <http://stackoverflow.com>
- <http://www.lastampa.it/>
- <http://it.wikipedia.org/>

- <http://www.uml.org/>
- <http://www.androidhive.info>
- <http://www.html.it/>
- <http://developer.android.com/>
- <http://json.org>