

Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica per il Management

**Progettazione e sviluppo di
un'applicazione iOS per il supporto
di strategie di mobile marketing
automation e di prossimità**

Relatore:
Chiar.mo Prof.
LUCIANO BONONI

Presentata da:
FRANCESCO MARISALDI

Co-relatore:
Ing. ALESSANDRO RIZZOLI

**Sessione II
Anno Accademico 2014/2015**

Indice

Introduzione	1
1 Cenni teorici	3
1.1 Mobile marketing	3
1.2 Marketing automation	4
1.3 Marketing di prossimità	5
2 Tecnologie utilizzate e stato dell'arte	7
2.1 La piattaforma MobileBridge	7
2.1.1 Il CMS	8
2.1.2 Il framework	10
2.2 Beacon	11
2.3 Stato dell'arte	12
3 Promotions: progettazione	15
3.1 Il Customer journey	15
3.2 Le User Stories	18
3.3 I requisiti funzionali	20
3.4 Mockup dell'applicazione	21
4 Promotions: sviluppo e scelte implementative	23
4.1 Model	24

4.2	AppDelegate	26
4.3	View e Controller	28
4.4	Scelte implementative	32
5	Promotions: simulazioni di utilizzo	35
5.1	Impostazione del CMS	36
5.2	Simulazioni di utilizzo	37
5.3	Presentazione dell'applicazione	41
6	Conclusioni	43
6.1	Possibili sviluppi	43
6.2	Considerazioni finali	44
	Bibliografia e Sitografia	47

Elenco delle figure

1.1	Share of U.S. Digital Media Time Spent By Platform	4
2.1	Esempio di processo gestito dalla piattaforma MobileBridge	8
2.2	Modello 3D di un beacon	11
3.1	Schema del Customer journey	16
3.2	Regione determinata da un beacon	18
3.3	Regione di un negozio	18
3.4	Area intorno ad un negozio	18
3.5	Schermata di benvenuto	21
3.6	Home dell'applicazione	21
3.7	Lista delle promozioni pubbliche	21
3.8	Schermata della promozione	22
3.9	Schermata dei negozi	22
3.10	Dettaglio del negozio	22
3.11	Schermata del wallet	22
3.12	Attivazione di una promozione	22
3.13	Profilo utente	22

4.1	L'icona dell'app	24
5.1	Schermata e notifica di benvenuto	37
5.2	Inserimento dei dati per il profilo	37
5.3	Condivisione dell'app	37
5.4	Notifica con promozione di benvenuto	38
5.5	La schermata con la promozione	38
5.6	Wallet con le promozioni salvate	38
5.7	I negozi della catena	38
5.8	Dettaglio del negozio selezionato	38
5.9	La schermata del profilo utente	38
5.10	Notifica all'uscita del negozio	39
5.11	Il sondaggio inviato con la notifica	39
5.12	Promozione per passaggi ripetuti	39
5.13	Notifica con promozione fedeltà	40
5.14	Il contenuto della promozione	40
5.15	Utilizzo della promozione fedeltà	40
5.16	Notifica in prossimità del negozio	40
5.17	La view di presentazione	40
5.18	Notifica per promozione benvenuto	40

Introduzione

Nel mercato globalizzato in cui le aziende si trovano a competere, uno dei fattori critici di successo con una sempre maggior rilevanza è certamente la strategia di marketing che l'azienda stessa attua; marketing inteso come l'insieme dei processi che portano alla definizione del valore per il cliente, alla generazione e comunicazione al cliente di una proposizione di valore irresistibile e a cogliere le opportunità che possano consentire di generare cash flow. Parallelamente, è indubbio che negli ultimi anni abbiamo assistito ad una trasformazione negli usi e costumi della società che ha portato lo smartphone ad essere uno degli strumenti più utilizzati da sempre più ampie fasce di popolazione, con la conseguente espansione della cosiddetta "app economy".

L'obiettivo della presente tesi è ideare, realizzare e presentare un'applicazione per dispositivi iOS, denominata "Promotions", legata ad un ipotetico brand con una catena di negozi, attraverso la quale sia possibile implementare alcuni casi d'uso relativi a particolari tipi di marketing che mettano al centro dell'esperienza utente l'utilizzo dell'app stessa, che diventa un vero e proprio canale e strumento di marketing, evidenziandone i potenziali effetti positivi e i risultati ottenuti.

Il progetto è stato realizzato durante un tirocinio nell'azienda GetConnected e fa uso della piattaforma di mobile marketing automation "MobileBridge". Il contesto preso in considerazione è quello di un brand che opera nel settore dell'alta moda (o in generale dell'abbigliamento), con una catena di retail store, che vuole proporre alla propria clientela promozioni, messaggi, sondaggi, sistemi di engagement e fidelity, attraverso appunto un'applicazione mobile. Il lavoro svolto è partito quindi dall'ideazione di un customer journey originale che comprendesse casi d'uso di mobile marketing automation e di prossimità con le

caratteristiche sopracitate: successivamente si è passati alla fase centrale dell'elaborato, cioè la progettazione e sviluppo dell'applicativo, concludendo poi con una simulazione di utilizzo.

La trattazione del progetto svolto parte da una breve descrizione e analisi del mobile marketing, del marketing automation e di prossimità, della piattaforma e delle tecnologie utilizzate. Prosegue con un'analisi sullo stato dell'arte di applicazioni e realtà simili a quella sviluppata già presenti sul mercato, per poi passare alla relazione sulla progettazione, sviluppo e realizzazione dell'app. Vengono poi evidenziate le scelte implementative attuate e descritte le simulazioni di utilizzo dell'app, con le conclusioni finali.

Capitolo 1

Cenni teorici

In questo capitolo si vogliono elencare e descrivere le tipologie di marketing attuate nell'elaborato, oltre a rendere esplicite, attraverso alcuni dati, le motivazioni che possono spingere un brand a sviluppare un'applicativo come quello trattato nella presente tesi.

1.1 Mobile marketing

L'ambito più generale in cui si pone l'applicazione sviluppata è quello del mobile marketing.

“Con l'espressione mobile marketing si intende ogni forma di avvicinamento a potenziali clienti attraverso qualche tipo di messaggio di marketing” attraverso i dispositivi mobili “E' particolarmente adatto a farvi trovare dai vostri clienti e a fare in modo che vengano a conoscenza dei vostri prodotti proprio nel momento in cui ne hanno bisogno. Comprende attività molto varie, come: mobile advertising, sms, applicazioni mobili, mobile site, social media” [1].

Il mobile marketing indica quindi l'insieme di quelle attività e strategie che, utilizzando lo smartphone o il tablet come canale di comunicazione, permettono di migliorare la comunicazione e l'iterazione con il proprio target, rafforzando la relazione cliente - marca; in questa tesi, esso viene attuato attraverso un unico strumento, l'applicazione sviluppata. Perché puntare sul mobile marketing, quindi

sul mondo mobile e in particolare sull'app economy? I dati e le tendenze a livello globale degli ultimi anni sono abbastanza chiari al riguardo.

Ad esempio, secondo un'indagine pubblicata da "comScore"ⁱ del giugno 2014 e riferita agli Stati Uniti, il 52% del tempo impiegato nella fruizione di media digitali avviene attraverso le mobile apps (vedi il grafico della Figura 1.1); l'utente medio possessore di uno smartphone scarica tre apps al mese; il 57% degli utenti usa una o più applicazioni ogni giorno nell'arco di un mese [2].

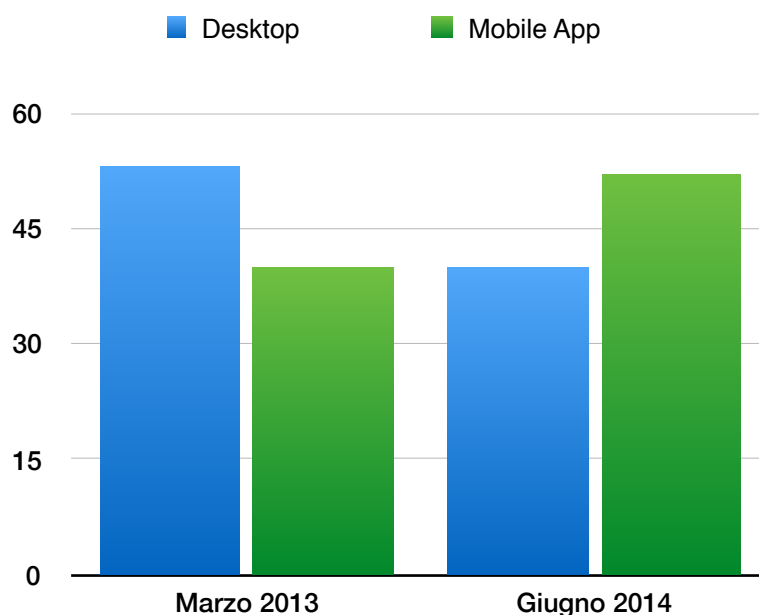


Figura 1.1: "Share of U.S. Digital Media Time Spent By Platform"

1.2 Marketing automation

“La marketing automation è quella tecnologia che permette di semplificare, automatizzare e misurare i compiti del marketing e i flussi di lavoro, così da poterne aumentare l'efficienza operativa e accrescere più rapidamente i ricavi”, più precisamente, “Con l'espressione Marketing automation definiamo il processo di utilizzo di un'unica piattaforma e di un'unica modalità per il monitoraggio dei KPIⁱⁱ, automatizzando tutte le attività di marketing personale e producendo i rapporti a chiusura di un ciclo, migliorando così l'efficacia di tutte le attività di marketing. La marketing automation si compone di tre parti che

ⁱ <http://www.comscore.com>

ⁱⁱ Key Performance Indicator, aiuta a definire e misurare i progressi compiuti per raggiungere gli obiettivi di una campagna o strategia di comunicazione.

devono: comprendere il percorso di un leadⁱ, attuare un processo automatico, consentire una reportistica di chiusura del ciclo”. I vantaggi generati dalla marketing automation possono essere molteplici:

- generare più lead
- fare di più con lo stesso team di marketing
- consolidare gli strumenti del marketing multicanale [3]

Riferendosi più specificamente al mondo mobile, il mobile marketing automation può riferirsi all’uso di strumenti software per eseguire, gestire e automatizzare processi e task del mobile marketing. Per esempio, chi gestisce app iOS o Android potrebbe voler automatizzare l’invio di notifiche push o messaggi in-app, segmentando l’invio al target di clienti da raggiungere [4].

Applicazioni di successo presenti in AppStore di iOS e nel Google Play Store, come “Subway Surfer” e “Plants vs Zombies” usano il mobile marketing automation per incrementare le proprie revenues, e grandi brand come Starbucks e ESPN lo usano per migliorare l’engagement nelle rispettive app [5]. Nel caso del presente progetto, la marketing automation avverrà, come spiegato in seguito, servendosi della piattaforma “MobileBridge”.

1.3 Marketing di prossimità

In generale, nell’ambito delle promozioni mobili, è determinante raggiungere i potenziali clienti nei momenti in cui è più probabile che siano disposti a ricevere un servizio: a questo proposito, i servizi georeferenziatiⁱⁱ sono sistemi che trasmettono segnali digitali ai dispositivi abilitati entro un raggio specifico [1].

Il marketing di prossimità, o proximity marketing, può essere indicato, con una definizione non rigorosa, come una tecnica di marketing che opera in un territorio circoscritto sfruttando tecnologie di comunicazione di tipo visuale e mobile al fine di promuovere la vendita di prodotti e servizi; non agisce su un target di utenti ben definito, ma sulle persone che si trovano in una determinata area e sono vicine a un dispositivo, apparato, sensore particolare, come ad esempio: RFID, NFC, motion capture, eye tracking, beacon [6].

Sempre in relazione al marketing di prossimità e georeferenziato, una tecnica di frequente utilizzo è quella del geo-fence, cioè recinto geografico, ossia la

ⁱ un lead è un contatto o cliente potenziale

ⁱⁱ location-based services, detti anche near-LBS

definizione di un perimetro virtuale per un'area geografica del mondo reale (ad esempio, il raggio attorno ad un negozio) sfruttata da applicazioni per fornire agli utenti servizi georeferenziati [7].

Il marketing location-based e di prossimità può comportare una serie di vantaggi:

- un target “catturato”: il cliente si trova già nei pressi della sede dell'attività;
- sviluppo di una relazione one-to-one;
- efficienza nelle spese di direct marketing: il materiale promozionale è mirato ed inviato in maniera digitale;
- gratificazione psicologica: il cliente sente una considerazione nei suoi confronti grazie alla comunicazione diretta, cosa che può aumentare la fedeltà alla marca;
- miglior ritorno sugli investimentiⁱ: è più probabile avere visite ripetute o un aumento degli acquisti per visita [1]

Secondo invece un rapporto di ricerca della società di analisi Berg Insightⁱⁱ, il valore totale del mercato pubblicitario mobile basato sulla localizzazione crescerà da 1,2 miliardi di euro nel 2013 a 10,7 miliardi di euro nel 2018, corrispondenti al 38% di tutta la pubblicità e marketing mobile. Il marketing e la pubblicità di prossimità rappresenteranno il 7% della pubblicità digitale e al 2% del totale della spesa pubblicitaria globale per tutti i media [8].

ⁱ ROI = Return On Investment

ⁱⁱ <http://www.berginsight.com>

Capitolo 2

Tecnologie utilizzate e stato dell'arte

2.1 La piattaforma MobileBridge

MobileBridge è un'azienda californiana fondata nel 2012, il cui core business è lo sviluppo di una tecnologia cloud software indirizzata al mondo enterprise, che ha portato alla creazione di una delle più avanzate piattaforme di mobile marketing automation: attraverso questa piattaforma i venditori possono creare, gestire e automatizzare customer journeys indirizzati al mondo mobile, migliorando l'engagement, il targeting, la fedeltà, il tasso di utilizzo e l'esperienza di utilizzo degli utenti all'interno dell'applicazione mobile, che può essere sviluppata da zero o già esistente. L'automazione è la caratteristica principale della piattaforma, a cui vengono affiancate funzionalità che permettono di implementare azioni di marketing di prossimità e geo-referenziato, e di avere un'analisi di utilizzo in real-time. La Figura 2.1 mostra un esempio di un processo gestibile grazie alla piattaforma¹.

I componenti utilizzati nel contesto del progetto sono fondamentalmente due: il Content Management System (CMS), un'interfaccia web attraverso la quale vengono gestiti tutti gli strumenti e le funzionalità offerte dalla piattaforma, e il

¹ da: <http://www.mobilebridge.com/solution.html>

framework, una libreria disponibile per sistemi iOS e Android, che permette l'utilizzo degli strumenti e funzionalità di MobileBridge all'interno di un'applicazione.

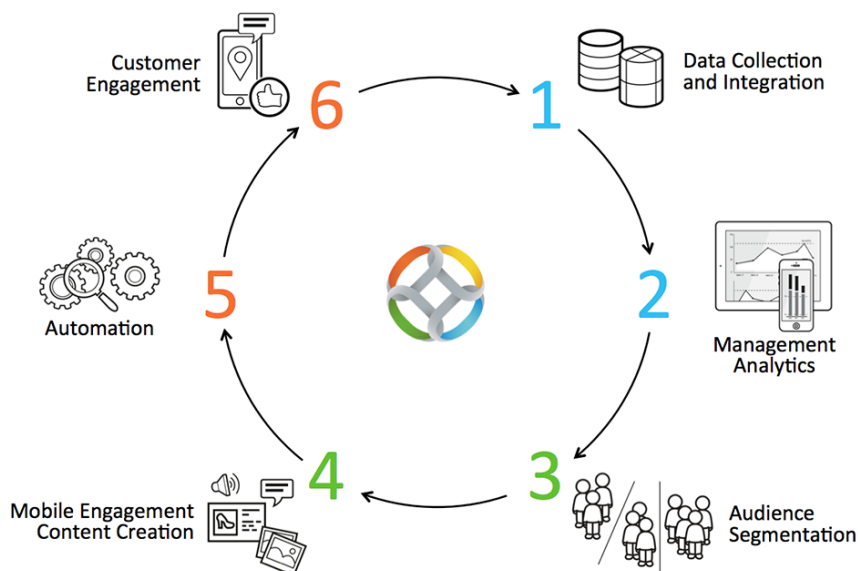


Figura 2.1: Esempio di processo gestito dalla piattaforma MobileBridge

Alcune considerazioni importanti: la piattaforma di MobileBridge è già in uso ad alcune aziende e partners, ma la versione del framework e del CMS utilizzata per lo svolgimento di questa tesi era, al momento dello sviluppo, una beta. Ciò ha significato l'utilizzo del framework con una documentazione per la maggior parte deprecata, dovendo quindi fare operazioni di "reverse ingeneering" e debug per comprendere il giusto uso di alcune funzioni, e soprattutto il dover lavorare con una piattaforma che non ha garantito il corretto funzionamento di alcune caratteristiche. In particolare, oltre a qualche raro problema nel sistema dei triggers descritto nel paragrafo successivo, non è stato possibile utilizzare il sistema di gestione dei beacons: per questo motivo, come si descriverà nel Capitolo 4 sullo sviluppo dell'app, si è deciso di gestire autonomamente, dove possibile, quei casi d'uso che prevedevano l'iterazione con un beacon.

2.1.1 Il CMS

Attraverso il CMS si possono gestire le quattro grandi aree della piattaforma che interessano ai fini della presente tesi:

- Gestione dell'audience e delle locations: si possono impostare le locations, cioè creare referenze agli store reali del brand, caratterizzandoli attraverso

l'indirizzo e un raggio espresso in metri che determinano il recinto geografico utilizzato dal framework per le azioni di geofencing, descrizione, immagine, telefono, mail, sito ufficiale, orari di apertura. Ad ogni location inoltre possono essere associati uno o più beacons, utilizzati per i casi d'uso di marketing di prossimità. Un'altra sezione fondamentale è quella del profilo utente: si possono cioè definire le caratteristiche (come età e sesso, le due pre-impostate e obbligatorie) che l'utente dovrà impostare, ovviamente attraverso l'app, per creare un proprio profilo. Esso permette di targettizzare in maniera molto profonda gli input da inviare ai clienti: infatti, le caratteristiche legate ad un profilo non sono solo quelle definite dall'utente, ma per ognuno la piattaforma tiene traccia ovviamente dei punti fedeltà accumulati e delle azioni intraprese dall'utente all'interno dell'applicazione. In sostanza, ad ogni profilo viene associata un'installazione dell'app: in questo modo posso sapere ad esempio quando è avvenuto l'ultimo utilizzo dell'app, se l'utente ha salvato una certa promozione, se è stato in una certa location, se ha interagito con un beacon, se ha risposto a un certo sondaggio eccetera. Infine, ad ogni profilo è associato un wallet e un inbox.

- Mobile engagement, promozioni e loyalty: è possibile creare e gestire promozioni, sondaggi, messaggi semplici e con contenuti multimediali, giochi promozionali, video. La creazione parte dagli elementi di base (titolo, descrizione) alla composizione dell'interfaccia grafica e dei contenuti attraverso un vero e proprio interface builder, e infine alla scelta delle impostazioni specifiche dell'oggetto. Ad esempio, per una promozione si deve determinare il valore dello sconto (o della carta fedeltà), il metodo di attivazione (barcode, codice numerico, qr-code), il numero totale delle promozioni disponibili e il numero di promozioni attivabili per ogni singolo utente, la data di scadenza, l'auto-salvataggio nel wallet degli utenti; per un sondaggio si possono definire le domande obbligatorie. Attraverso il CMS è anche possibile definire le regole sul sistema di fidelity da legare all'utilizzo dell'app. Maggiori dettagli sono forniti nella descrizione sulla simulazione di utilizzo dell'app, nel capitolo 5.
- Marketing automation: raccoglie le funzionalità chiave della piattaforma. Un'automation si può descrivere nel seguente modo: in base ad un prestabilito evento scatenante (trigger) viene inviato un determinato oggetto ad un determinato target di utenti, con delle impostazioni aggiuntive da poter specificare. Il trigger può essere scelto da una serie molto ampia di eventi: iterazione (entrata o uscita) con un beacon o con una posizione geografica, triggers temporali, esterni o legati ad azioni di condivisione social, azioni legate all'utilizzo dell'app come prima installazione, visualizzazione o utilizzo di un

determinata promozione o messaggio, raggiungimento di una certa quantità di punti fedeltà. Una volta scelto il trigger, si definisce l'oggetto dell'automation, che può essere una promozione, un sondaggio, un messaggio, eccetera. A questo punto può essere impostato il target a cui inviare l'oggetto, definito creando uno o più profili che hanno come caratteristiche quelle raccolte dalla piattaforma nella profilazione degli utenti (vedi gestione dell'audience): quindi, ad esempio, posso targettizzare l'invio agli utenti che hanno una certa età, che sono entrati in una certa location o che sono passati nelle vicinanze di una location, che hanno eseguito una certa funzione all'interno dell'app tracciata dal framework, eccetera. Infine, le impostazioni da poter scegliere sono principalmente: le modalità di invio, che possono essere una notifica push o il salvataggio nell'inbox o wallet dell'utente, il periodo di validità, il numero di volte che l'automation può essere attivata (in totale e dal singolo utente). Una volta salvata, l'automation, che può essere caratterizzata da un titolo e una descrizione, diventa già attiva. Oltre alle automation, si possono inviare notifiche push o elementi nell'inbox degli utenti limitando l'automatizzazione alla definizione del target, al periodo di validità e ad una eventuale ripetizione della stessa notifica in un lasso temporale.

- Analytics: una sezione del CMS è dedicata alla visualizzazione dei dati raccolti per i diversi ambiti della piattaforma. E' possibile vedere le statistiche sull'utilizzo dell'app (installazioni, sessioni, utenti attivi, azioni tracciate), i dati di accesso alle locations e le statistiche relative all'iterazioni con i beacon, i risultati dei sondaggi sottoposti, i dati sulle promozioni inviate (numero di salvataggi e utilizzi), un'analisi demografica in base ai profili creati.

2.1.2 Il framework

Passando al framework, l'attivazione delle sue funzionalità può avvenire o all'interno dell'applicazione, attraverso ad esempio un'azione dell'utente, o sfruttando le caratteristiche del sistema operativo iOS, quali ad esempio le notifiche push. Per implementare le funzionalità offerte dal framework, sono disponibili un'ampia collezione di APIs, classificabili in tre aree: costruttori, delegates, metodi pubblici. Per una descrizione dettagliata delle API utilizzate fare riferimento al capitolo 4 sullo sviluppo dell'app.

2.2 Beacon

Oltre alla piattaforma utilizzata, si fornisce una descrizione di che cos'è un beacon e della tecnologia iBeacon, vista la gestione diretta nel codice sviluppato dei casi d'uso riguardanti questa tecnologia a causa dei problemi descritti nel paragrafo precedente.

Un beacon è un piccolo dispositivo wireless (Figura 2.2) basato sullo standard Bluetooth Low Energy® che trasmette segnali radio continui chiamati “advertising packets”; questi segnali possono essere ricevuti da qualsiasi smartphone equipaggiato con Bluetooth 4.0 e che sia nel range del beacon trasmettitore, senza che sia necessario il pairing fra i due. Quando lo smartphone riceve l'advertising packet, la distanza e la posizione relativa dal beacon possono essere calcolate senza l'ausilio di ulteriori sistemi (come il GPS); ogni beacon è caratterizzato da un id univoco trasmesso all'interno dell'advertising packet. Il range massimo del segnale dipende dal luogo e dalla transmission powerⁱ, che può essere settata e in linea teorica può arrivare fino a 70 metri circa, così come può essere settato “l'advertising interval”ⁱⁱ. Le situazioni di utilizzo che si possono creare utilizzando un beacon sono le più svariate: ad esempio, possono essere caricati dei contenuti da Internet quando lo smartphone si avvicina ad un determinato beacon [9].



Figura 2.2: Modello 3D di un beacon

iBeacon è una tecnologia introdotta da Apple con l'uscita di iOS 7 (e su dispositivi con esso compatibili dotati di Bluetooth 4.0) che definisce alcuni standard che il beacon (dispositivo trasmettente) deve rispettare per poter comunicare con iPhone, iPod Touch e iPad, oltre a funzionalità a livello software. Le specifiche principali stabiliscono che i componenti chiave dell'advertising packet trasmesso devono essere un proximity UUIDⁱⁱⁱ e due valori interi compresi fra 1 e 65535, chiamati major e minor. UUID, major e minor identificano in maniera univoca un iBeacon. Un altro elemento fondamentale dell'advertising packet è l'ultimo byte, cioè il pacchetto utilizzato per calcolare la distanza

ⁱ la potenza della trasmissione

ⁱⁱ l'intervallo di trasmissione

ⁱⁱⁱ Universally Unique Identifier

dall'iBeacon. Inoltre, il beacon deve essere in modalità “non-connectable”, cioè i suoi parametri non possono essere modificati dagli utenti; l'advertising interval consigliato è 100ms, la transmission power a -12 dBm. Sfruttando le APIs introdotte nel framework “CoreLocation”, un'applicazione può, ad esempio, monitorare l'entrata o l'uscita del dispositivo dalla regione identificata dal beacon, o determinare una stima della prossimità del dispositivo al beacon, classificandola in 4 diversi stati: immediate, near, far e unknown [10].

Il beacon utilizzato durante la realizzazione di questo progetto è stato prodotto dall'azienda kontakt.ioⁱ, che fornisce dispositivi già compatibili allo standard iBeacon, oltre ad una app da cui poter operare modifiche alle impostazioni. L'advertising interval è settato di default a 350 ms e la potenza di trasmissione a -12 dBm: queste impostazioni garantiscono un range operativo di circa 20 metri e un'autonomia fino a 2 anni.

2.3 Stato dell'arte

In questo paragrafo vengono elencate e descritte alcune applicazioni già presenti sul mercato con finalità in parte simili a quella realizzata, oltre a piattaforme di mobile marketing automation simili a quella di MobileBridge. Si precisa che la ricerca è stata effettuata per proporre solo alcuni esempi di realtà esistenti e non per offrire uno spaccato completo del mercato preso in considerazione, in quanto non è ovviamente lo scopo della presente tesi.

Per quanto riguarda le piattaforme rivolte al mobile marketing automation e simili a quella utilizzata, ne sono state trovate due:

- Artisanⁱⁱ: piattaforma che permette ad aziende e sviluppatori di creare relazioni più profonde con gli utenti attraverso il mobile marketing automation. Le principali caratteristiche sono: notifiche push e messaggi in-app targettizzati, track degli eventi automatico e omni-channel, segmentazione in base al comportamento degli utenti, integrazione per sistemi iOS, Android e per web apps.
- TapTargetⁱⁱⁱ: anche questa piattaforma si propone come strumento per il mobile marketing automation. Le caratteristiche più importanti sono: targeting in base alla localizzazione dell'utente utilizzando geofences e beacons,

ⁱ <https://support.kontakt.io/hc/en-gb>

ⁱⁱ <http://www.useartisan.com>

ⁱⁱⁱ <http://www.taptarget.com>

profilazione degli utenti, notifiche push, builder completo per creare campagne e comunicazioni in ambito mobile, semplice integrazione attraverso l'SDK per sistemi iOS e Android.

Fra le applicazioni che attuano strategie di marketing di prossimità, un esempio lampante e di successo è sicuramente l'italiana "CheckBonus", un'applicazione che funziona come una carta fedeltà, consente di accumulare punti e con essi ottenere premi e promozioni, entrando nei punti vendita delle catene convenzionate (ad esempio MarcoPolo Expert, Adidas e Coin). Per i clienti è un sistema che permette di accumulare punti velocemente e ottenere premi utilizzando lo smartphone come "carta fedeltà universale" per diversi store. Per le catene di venditori è un generatore di store traffic e una piattaforma di marketing: non appena l'utente entra nel punto vendita, CheckBonus si trasforma in uno strumento di proximity marketing che premia l'utente quando interagisce con il punto di vendita e i prodotti al suo interno. Inoltre l'app accompagna l'utente in tutto il percorso dell'acquisto, fuori e dentro lo store, utilizzando tutte le tecnologie necessarie al raggiungimento di questo obiettivo: dalle notifiche geofencing ad appunto il proximity marketing in store tramite iBeacon [8].

La filosofia su cui si basa CheckBonus è ispirata a quella che ha portato al successo su larga scala negli Stati Uniti "shopkick", che al 27 novembre 2015 fa registrare carte regalo e premi utilizzati attraverso l'applicazione per un valore totale di circa 63 milioni di dollariⁱ.

Infine, fra le aziende italiane che offrono soluzioni di marketing di prossimità come principale business si possono citare JoinTag, Open Service e Makeitapp.

ⁱ <https://www.shopkick.com>

Capitolo 3

Promotions: progettazione

3.1 Il Customer journey

La fase di progettazione è partita dall'ideazione del customer journey che si vuole attuare attraverso l'applicazione "Promotions". Il customer journey, ossia il percorso del cliente, è "il modo in cui un potenziale cliente si comporta per arrivare a completare un acquisto. L'insieme di questi momenti, tra loro differenti, crea la combinazione che permette di far arrivare l'utente dove noi desideriamo" [3].

Questa prima fase riguarda gli obiettivi e scenari di marketing che si vogliono raggiungere e realizzare. Come descritto, il contesto preso in considerazione è quello di un brand che opera nel settore dell'alta moda, con una catena di retail store, che vuole proporre alla propria clientela promozioni, messaggi, sondaggi, sistemi di engagement e fidelity. Lo schema discusso, raffigurato in Figura 3.1, non include tutto il percorso decisionale e le azioni che il cliente compie nelle diverse fasi d'acquisto e non specifica ovviamente i dettagli tecnici dei casi d'uso, ma si limita a descrivere quegli scenari che possono prevedere l'utilizzo dell'applicazione da parte del cliente, per raggiungere determinati obiettivi.

Il punto d'inizio del customer journey è il download e l'avvio dell'applicazione da parte dell'utente: l'informazione dell'esistenza dell'app e una sua descrizione può essere reperita dal cliente in vari modi, ad esempio sul sito ufficiale del

brand, attraverso una mail promozionale, direttamente all'interno di un negozio tramite un QR-code, attraverso i canali social del brand.

Una volta avviata l'applicazione, viene immediatamente notificata all'utente l'assegnazione di una certa quantità di punti fedeltà al suo profilo, che viene creato contestualmente al primo avvio. L'utente deve poi inserire i propri dati per completare il profilo a lui assegnato: ovviamente nel caso l'utente non inserisca i propri dati, non può effettuare ulteriori operazioni, e quindi lo scenario proposto nel customer journey termina. Una volta completato il profilo, l'utente può decidere se effettuare una condivisione su un social network dell'applicazione: se acconsente e la condivisione va a buon fine, gli verrà notificata l'assegnazione di una speciale promozione di benvenuto, utilizzabile recandosi ad uno store (nel caso il non sia già all'interno). Sia che il cliente usi la promozione o faccia comunque degli acquisti, sia in caso contrario, una volta uscito dal negozio gli viene proposto un sondaggio sull'esperienza avuta nello store.

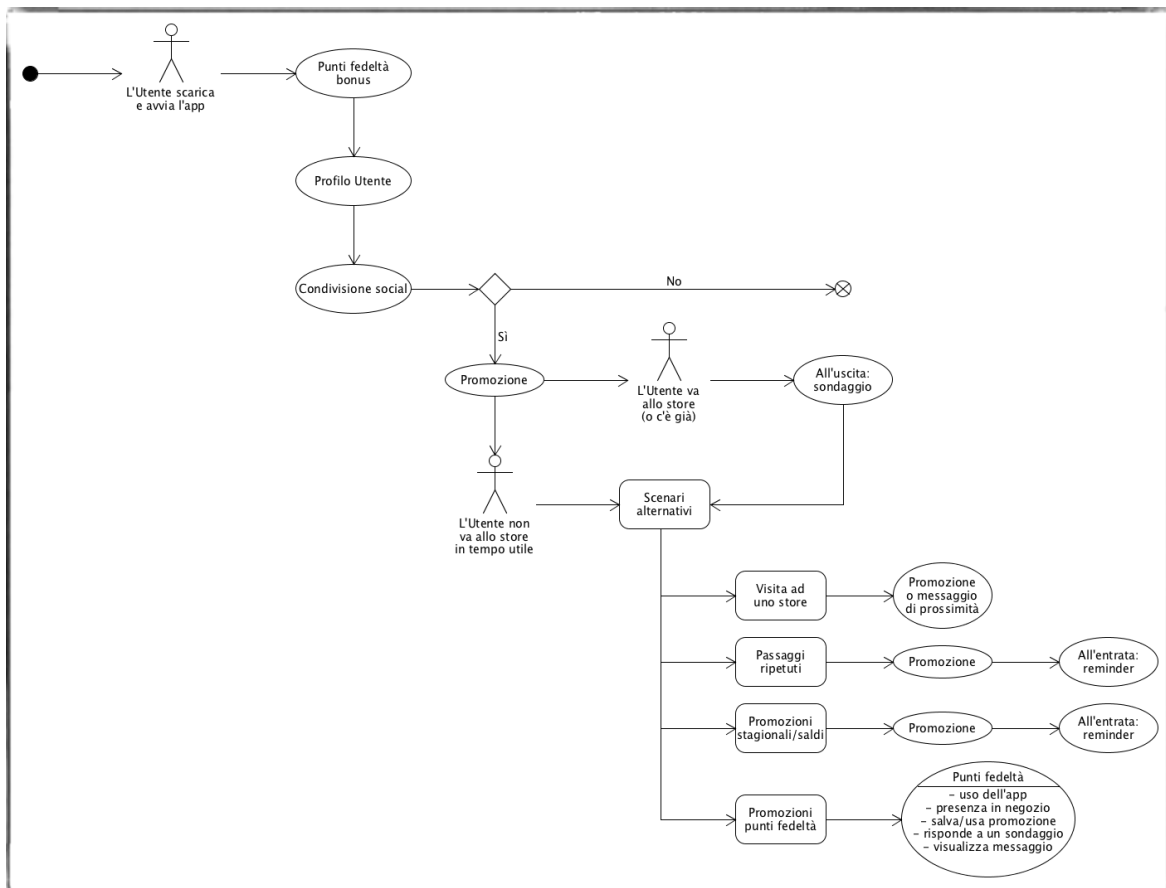


Figura 3.1: Schema del Customer journey

Nel caso in cui l'utente non effettui la condivisione (non ricevendo quindi il coupon), oppure non si rechi ad uno store per usare la promozione ricevuta, o

comunque in un tempo successivo all'utilizzo della promozione di benvenuto stessa, sono stati ideati diversi scenari alternativi che coinvolgano sempre l'utilizzo dell'applicazione. Nel caso in cui, ad esempio, l'utente passi ripetutamente nelle vicinanze di un negozio senza entrarci (per due o tre volte in un raggio di 100 metri ad esempio), inviargli una promozione potrebbe facilitarne l'engagement. Oppure si possono pianificare notifiche per promozioni in base alla stagione, ai saldi o particolari eventi in arrivo. Un altro caso previsto è la possibilità di notificare promozioni o messaggi speciali all'arrivo del cliente in uno store; sempre legato alla prossimità di uno store, potrebbe essere notificata solo una volta per utente, alla prima visita, la presentazione del negozio stesso. Infine ci sono le casistiche che regolano il sistema di fidelity legato all'applicazione: oltre ai punti di benvenuto, si può incentivare l'utilizzo dell'app con punti fedeltà all'apertura della stessa, al salvataggio e uso di promozioni, alla visualizzazione di messaggi, alla risposta ad un sondaggio proposto, alla presenza in uno dei negozi della catena. Al raggiungimento di una certa quantità di punti fedeltà, viene automaticamente proposta una promozione, che va a scalare una parte dei punti stessi.

Il customer journey ideato, quindi, prova a mettere in atto diverse strategie di marketing: l'automazione nella proposta di promozioni, alcuni eventi che scaturiscono dall'utilizzo dell'applicazione stessa o dalla presenza in prossimità degli store del brand, e il sistema di fidelity, anch'esso automatizzato, che porta l'utente ad un utilizzo maggiore e continuato nel tempo dell'app, cercando di aumentare la "brand awareness" e la fedeltà del cliente. Inoltre non è specificato nello schema, ma è possibile un forte targeting delle promozioni e messaggi proposti grazie alla condivisione di alcuni dati del profilo (sesso ed età) o alla prossimità dell'utente a determinati reciti geografici o sensori: una segmentazione che aumenta certamente l'efficacia delle comunicazioni proposte attraverso l'app. Automazione e marketing di prossimità, coadiuvati dalla targettizzazione e dal sistema di fidelity, sono quindi i principali strumenti di mobile marketing che l'applicazione dovrà essere in grado di garantire e implementare. Ovviamente, le situazioni ideate sono esemplificative delle innumerevoli possibilità che potrebbero essere attuate: il numero dei casi d'uso è limitato per permetterne poi l'effettiva verifica.

Come descritto nel Capitolo 2, la piattaforma di Mobile Bridge garantisce la gestione delle attività riguardanti il marketing automation, il sistema di fidelity, il targeting e parte del marketing di prossimità; l'ambito e l'obiettivo della tesi è infatti quello di ideare il customer journey presentato, pensato in modo da essere realizzabile sfruttando questa piattaforma, e lavorare non sul "backend" ma,

come verrà descritto nel prosieguo, sulla progettazione e sviluppo dell'applicazione utente, con l'obiettivo finale di proporre un applicativo che possa offrire ad un ipotetico brand e ai suoi clienti i vantaggi e le opportunità descritte.

I tre diversi scenari che riguardano casi d'uso di marketing di prossimità previsti dal customer journey sono raffigurate nelle Figure 3.2, 3.3 e 3.4.

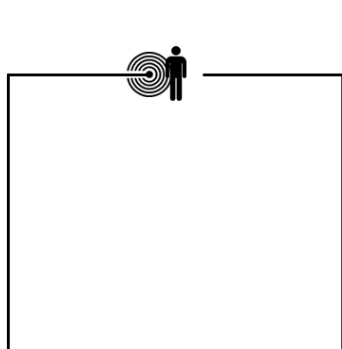


Figura 3.2: Regione determinata da un beacon

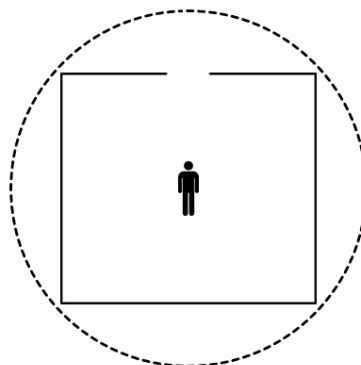


Figura 3.3: Regione di un negozio

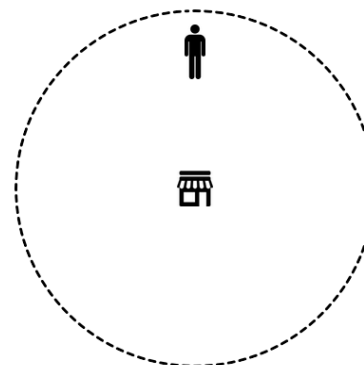


Figura 3.4: Area intorno ad un negozio

3.2 Le User Stories

Una volta definito il customer journey da implementare, è iniziata la progettazione dell'app Promotions, seguendo questo percorso: sono partito dalla scrittura delle user stories, per poi stabilire i requisiti funzionali e infine realizzare alcuni mockups della user interface. Le user stories e i requisiti funzionali da rispettare non sono stati decisi in maniera arbitraria, ma sono stati concordati con il tutor aziendale e definiti anche in base ai casi d'uso previsti dal customer journey.

Una User Story è la definizione ad alto livello di una funzionalità, scritta nella forma: *“As a [role] I want [something] so that [benefit]”* [11]

Le user stories definite sono le seguenti:

- 1) Struttura app & Integrazione Mobile Bridge: come Venditore, voglio avere un'applicazione per iPhone organizzata in quattro tab in grado di interfacciarsi con la piattaforma di MobileBridge, di sfruttare le funzionalità presenti nell'SDK, di ricevere e gestire notifiche push, per poter implementare i casi d'uso previsti.

- 2) Primo utilizzo: come Utente, voglio avere la possibilità di inserire le mie informazioni di base (nome, città, genere, data di nascita) per creare il mio profilo. Successivamente, devo poter scegliere fra andare direttamente alla home dell'applicazione o effettuare le azioni proposte dall'applicazione per avere una promozione di benvenuto.
- 3) Promozione di benvenuto: come Venditore, voglio proporre una promozione di benvenuto all'utente che, dopo aver scaricato l'app e aver completato il proprio profilo, effettui una condivisione dell'applicazione su un social network.
- 4) Visualizzazione delle promozioni: come Utente, vorrei visualizzare le promozioni disponibili, con la possibilità per ognuna di averne i dettagli e salvarla nel Wallet per un successivo utilizzo.
- 5) Visualizzazione sondaggio o messaggio: come Utente, voglio visualizzare e interagire con un sondaggio o un messaggio in una schermata specifica, all'arrivo di una notifica push o dall'inbox.
- 6) Lista dei negozi: come Utente, vorrei consultare, in una lista, l'elenco dei negozi della catena, con alcune informazioni dettagliate (indirizzo, telefono, mail, sito web) da poter utilizzare e una mappa con la posizione di ciascun negozio da poter visualizzare.
- 7) Wallet: come Utente, vorrei consultare in una sezione apposita le promozioni salvate, divise per attive, usate e scadute, e consultare una promozione selezionata, avendo la possibilità di utilizzare quelle attive.
- 8) Profilo utente: come Utente, voglio avere una sezione dell'app nella quale visualizzare e modificare le informazioni sul mio profilo, visualizzare la situazione dei miei punti fedeltà, gestire le preferenze.
- 9) Inbox: come Utente, voglio consultare le notifiche arrivate in una sezione dell'app (escluse quelle cancellate automaticamente dal sistema), con la possibilità di cancellarle tutte o interagire con esse (visualizzare una promozione e salvarla, rispondere ad un sondaggio, visualizzare un messaggio).
- 10) Promozioni: come Venditore, voglio proporre, attraverso una notifica, delle promozioni, sondaggi o messaggi agli utenti dell'app in base a diversi fattori o eventi, e voglio definire alcune azioni e regole standard in base alle quali vengono assegnati dei punti fedeltà all'utente.

3.3 I requisiti funzionali

Di seguito sono riportati invece i requisiti funzionali più importanti che l'applicazione "Promotions" dovrà rispettare e garantire al termine della fase di sviluppo:

- l'applicazione deve poter essere eseguita su dispositivi iPhone e iPod Touch con sistemi operativi iOS 8 e successivi, non è invece prevista una ottimizzazione per iPad;
- l'interfaccia grafica deve essere ottimizzata per le varie dimensioni dei display dei dispositivi supportati e deve essere adeguata all'utilizzo finale da parte degli utenti;
- l'applicazione deve garantire l'integrazione con il framework di Mobile Bridge e delle funzionalità dello stesso elencate più avanti;
- la struttura dell'app deve basarsi su un TabBarController con 4 tab: Home, Negozi, Wallet e Profilo;
- al primo avvio deve essere gestita la creazione del profilo dell'utente ed una facoltativa condivisione social;
- l'applicazione deve gestire il salvataggio e l'aggiornamento in locale dei dati del profilo dell'utente, oltre al salvataggio e l'aggiornamento online di quei dati condivisi con il CMS di MB;
- contenuti e funzionalità della sezione Home: informazioni sull'app, schermata per i banner promozionali, collegamento e lista delle promozioni pubbliche disponibili con possibilità di visualizzazione e aggiornamento, riepilogo del profilo utente;
- contenuti e funzionalità della sezione Negozi: elenco dei negozi, dettaglio per ogni negozio con indirizzo, mappa e azioni correlate (telefono, mail, percorso, sito web);
- contenuti e funzionalità della sezione Wallet: elenco delle promozioni salvate dall'utente divise per attive, usate e scadute, con possibilità di visualizzazione di ogni singola promozione, di eliminazione e aggiornamento;
- contenuti e funzionalità della sezione Profilo: riepilogo dei dati dell'utente con possibilità di effettuare e salvare modifiche, visualizzazione punti fedeltà, gestione delle preferenze e visualizzazione dell'inbox;

- l'applicazione deve essere in grado di ricevere e gestire notifiche push e notifiche locali in tre diversi scenari: applicazione non in esecuzione, applicazione in uso, applicazione aperta ma non in uso (ad esempio in background);
- l'applicazione deve gestire la visualizzazione delle schermate (gestite dal framework di MB) di promozioni, sondaggi (di due tipologie) e messaggi (di due tipologie) all'arrivo di una notifica push o dall'inbox.

3.4 Mockup dell'applicazione

Successivamente alla scrittura delle user stories e alla definizione dei requisiti funzionali, ho realizzato alcuni mockups dell'UI dell'applicazione, utili per avere un'idea generale sul lavoro da fare in fase di sviluppo sull'interfaccia grafica e per avere una base concreta e visibile dell'applicazione fin da questa fase iniziale.

Il software utilizzato per realizzare i mockups è "Balsamiq Mockups 3"ⁱ nella versione di prova di 30 giorni. Nelle Figure 3.5 - 3.13 riporto quindi i mockups realizzati, che riguardano le schermate principali dell'applicazione.



Figura 3.5: Schermata di benvenuto



Figura 3.6: Home dell'applicazione



Figura 3.7: Lista delle promozioni pubbliche

ⁱ <https://www.balsamiq.com/products/mockups/>



Figura 3.8: Schermata della promozione



Figura 3.9: Schermata dei negozi

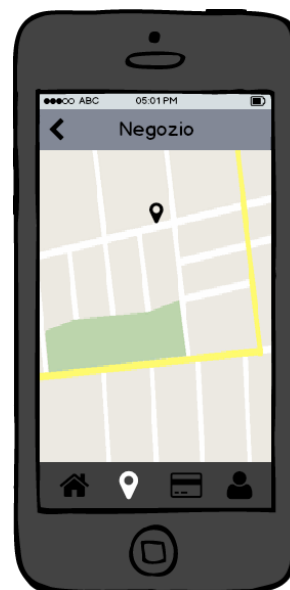


Figura 3.10: Dettaglio del negozio



Figura 3.11: Schermata del wallet



Figura 3.12: Attivazione di una promozione



Figura 3.13: Profilo utente

Capitolo 4

Promotions: sviluppo e scelte implementative

Una volta terminata la fase di progettazione, sono passato allo sviluppo del codice dell'applicazione, la parte centrale della tesi. Il metodo seguito è stato il seguente: partendo dalle user stories, ho definito per ognuna una serie di task della durata di 2 o 4 ore ciascuno che descrivessero, a grandi linee, le attività da fare e implementare per raggiungere gli obiettivi descritti in precedenza ad alto livello. In questo modo ho potuto stimare in anticipo il tempo necessario allo sviluppo, che si è rilevato sostanzialmente quello previsto, circa cento ore.

Il linguaggio di programmazione utilizzato è stato l'Objective-C, utilizzando Xcode come IDE. Il pattern di programmazione che ho cercato il più possibile di utilizzare è quello fortemente consigliato per la programmazione di applicazioni iOS, ossia il Model View Controller. Il "deployment target" scelto, quindi la versione del sistema operativo minima richiesta, è iOS 8: seguendo le specifiche, è stato scelto di supportare solo dispositivi iPhone (e iPod Touch), nella sola modalità portrait.

Il framework più importante utilizzato è ovviamente quello di MobileBridge, oltre a tutti i frameworks da cui esso dipende. Per il download asincrono di

immagini con cache di supporto ho utilizzato la libreria “SDWebImage” sviluppata da Olivier Poitreyⁱ.

Per quanto riguarda l’aspetto grafico e il layout dell’applicazione, le views sono state gestite per la maggior parte attraverso l’interface builder integrato in Xcode, utilizzando il sistema dell’autolayout per adattare i contenuti alle varie dimensioni dei display. Inoltre, nella descrizione del codice sviluppato che segue, si farà riferimento al codice, le classi e i metodi ritenuti più importanti.



Figura 4.1:
L'icona dell'app

4.1 Model

La descrizione del codice sviluppato parte dal “model” dell’applicazione, cioè l’insieme di classi che definiscono gli oggetti e il loro comportamento, utilizzati nel resto del progetto. Tutte le classi descritte sono sottoclassi di NSObject:

- User [name, birthday, gender, city, points]: è l’oggetto che rappresenta l’utente che usa l’applicazione, caratterizzato dai dati inseriti al primo avvio e modificabili, in particolare nome, data di nascita, genere, città, oltre ai punti fedeltà recuperati dinamicamente dal framework in alcuni controllers. Ha un metodo di inizializzazione personalizzato; inoltre, per assicurarne la consistenza tra diversi utilizzi dell’app, l’oggetto viene salvato e caricato in NSUserDefaults grazie al fatto che la classe aderisce al protocollo NSCoder attraverso l’implementazione di due metodi:
 - (void)encodeWithCoder:(NSCoder *)encoder
 - (id)initWithCoder:(NSCoder *)decoder
- Setting [title, status, tag]: è l’oggetto che rappresenta le preferenze impostabili dalla sezione “Profilo”. Ogni impostazione è caratterizzata da un titolo, uno status booleano e un tag che sarà associato, nel controller del profilo, al corrispondente UISwitch, per capire, in caso di modifiche, quale viene utilizzato. Ha inoltre un metodo di inizializzazione personalizzato.
- Poll [ID, title, type]: è l’oggetto che rappresenta il tipo di sondaggio più semplice, caratterizzato da un id, stringa utilizzata dal framework per richiedere il contenuto corretto della view, un titolo e un tipo, utile a capire se il poll viene inizializzato da una notifica o da un banner. Ha un metodo che ne permette

ⁱ <https://github.com/rs/SDWebImage> e <http://hackemist.com/SDWebImage/doc/>

l'inizializzazione a partire dai dati contenuti in un NSDictionary, utilizzato nel caso di arrivo di una notifica push o di richiesta di apertura dall'inbox:

- *(instancetype)initWithDictionary:(NSDictionary *)dictionary*

- Promotion [ID, title, desc, type, endDate, imageName, imageUrl]: è la classe che rappresenta l'oggetto promozione, caratterizzato da un id, un titolo, una descrizione, un tipo, una data di scadenza, url e nome dell'immagine rappresentante la promozione. L'oggetto istanziato verrà poi utilizzato per creare la view della promozione attraverso il costruttore fornito dal framework MB, per cui sono determinanti le properties "ID" e "type". Ha due diversi metodi che ne permettono l'inizializzazione da un NSDictionary.
- RichMessage [ID, title, type]: è l'oggetto che rappresenta il messaggio con contenuti multimediali e formattato, caratterizzato esattamente come il Poll, quindi da un id, un titolo e tipo. Ha un metodo che ne permette l'inizializzazione a partire dai dati contenuti in un NSDictionary.
- Survey [ID, title, type]: è l'oggetto che rappresenta il tipo di sondaggio più complesso, caratterizzato anch'esso come il Poll, quindi da un id, un titolo e tipo. Ha un metodo che ne permette l'inizializzazione a partire dai dati contenuti in un NSDictionary. Si noti che la creazione di tre oggetti simili ma distinti, Poll, RichMessage e Survey, si è resa necessaria in quanto il framework definisce tre distinti tipi, PollViewFromType, RichMessageViewFromType e SurveyViewFromType, da associare alla property "type".
- WalletItem [ID, title, desc, type, endDate, imageName, imageUrl, promotionMaxRedeeming, promotionRedeemed, activationCode]: rappresenta l'oggetto contenuto nel wallet, che è collegato al profilo nella piattaforma di MB. E' quindi sostanzialmente una promozione salvata, con alcune property aggiuntive: un eventuale codice di attivazione, il numero massimo di attivazioni consentite per utente e il numero delle attivazioni effettuate. Si noti che la property "type" è concettualmente diversa da quella del model "Promotion": in questo caso, il tipo corrisponde allo stato della promozione salvata nel wallet, che può essere attiva, usata o scaduta. Ha un metodo che ne permette l'inizializzazione a partire dai dati contenuti in un NSDictionary.

L'unica classe facente parte del model dell'applicazione gestita direttamente dal framework MB e quindi non sviluppata nel corso della realizzazione del progetto è "MB_LocationType", classe che rappresenta l'oggetto negozio, caratterizzata principalmente da id, titolo, indirizzo, telefono, email, url del sito web, latitudine e longitudine.

4.2 AppDelegate

Una descrizione particolare va fatta per l'AppDelegate. Oltre ai ruoli standard di questa classe, come la definizione del codice eseguito all'avvio dell'app, le risposte ai cambiamenti di stato dell'app e agli eventi che la coinvolgono, all'interno dell'AppDelegate vengono gestite altre quattro fondamentali funzionalità: il framework MobileBridge, l'inizializzazione dell'utente, la ricezione di notifiche push e locali, la gestione autonoma della regione determinata da un iBeacon.

Per utilizzare il framework di MobileBridge, vanno dichiarati un oggetto MobileBridgeLibrary, chiamato *mbLib*, e il delegate "MobileBridgeLibrary-Delegate" nell'AppDelegate, come suggerito nella documentazione ufficiale. L'inizializzazione avviene all'interno del metodo:

```
- (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
```

utilizzando le chiavi fornite dalla piattaforma, insieme al resto del codice di inizializzazione dell'app. Il metodo statico per creare un handler all'istanza della libreria è:

```
+ (id)initLibrary:(NSString*)clientKey clientSecret:(NSString*)clientSecret
```

Ogni volta che qualsiasi altro controller vorrà utilizzare un metodo pubblico o un costruttore di un oggetto della libreria, dovrà richiamare, attraverso l'AppDelegate, l'handler creato, *mbLib*, dichiarando inoltre ulteriori delegates qualora richiesto: l'AppDelegate è quindi l'unico "punto d'accesso" alle funzionalità offerte dal framework MobileBridge. Il delegate dichiarato è richiesto dalla libreria per l'invio di messaggi di risposta all'applicazione. L'AppDelegate si occupa anche di tracciare il tempo di utilizzo dell'applicazione utilizzando i metodi della libreria MB per avviare il tracciamento, quando l'app diventa attiva, e stopparlo, quando l'app entra in background o viene terminata.

Come spiegato nel Capitolo 2, si è resa necessaria, per quanto possibile, la gestione autonoma di quei casi d'uso che richiedono l'iterazione dell'app con un dispositivo iBeacon. Anche l'implementazione di questa funzionalità è gestita all'interno dell'AppDelegate, sfruttando il framework CoreLocation dell'SDK di iOS. Vengono istanziati due oggetti delle classi CLLocationManager e CLBeaconRegion, oltre a definire l'AppDelegate come "CLLocationManager-Delegate": il *location manager* permette di stabilire i parametri che determinano

quando inviare eventi location-based, mentre la *beacon region* definisce il tipo di regione, basata sulla prossimità del device ad un beacon bluetooth [12]. La modalità con la quale un beacon “segnala” la propria presenza e la propria identità sono state descritte nel Capitolo 2. Nel codice, vengono quindi definiti l’UUID e l’identifier del *CLBeaconRegion*; il monitoraggio viene fatto partire, insieme al codice di inizializzazione dell’app, utilizzando il metodo fornito in questa maniera:

```
[_locationManager startMonitoringForRegion:_beaconRegion];
```

Attraverso i metodi forniti dall’SDK, nel momento in cui il location manager determina che ci si trova all’interno della regione dichiarata, si eseguono due diverse azioni in base alla prossimità del beacon individuato: nel caso in cui la prossimità rilevata sia di tipo *CLProximityNear* viene inviata una notifica locale, la cui apertura da parte dell’utente comporta l’istanziamento della view di presentazione del negozio. Nel caso invece che la prossimità sia di tipo *CLProximityImmediate*, viene utilizzato il metodo della libreria *MobileBridge* per segnalare l’evento alla piattaforma, responsabile a sua volta dell’invio di una notifica push.

Per quanto riguarda l’inizializzazione e gestione dell’utente, viene definito nell’*AppDelegate* un oggetto pubblico di tipo “User”, *currentUser*: è questa istanza della classe *User* che verrà utilizzata da tutte le altre classi del progetto per leggere o aggiornare le informazioni relative all’utente. Nel metodo *loadUser*, al primo avvio vengono assegnati agli attributi di *currentUser* valori di default; per tutti gli altri avvii, *currentUser* verrà inizializzato partendo dall’istanza salvata nell’archivio *NSUserDefaults*. Il sistema funziona perché, sempre nell’*AppDelegate*, viene aggiunto il seguente observer al notification center:

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(currentUserUpdateHandler)  
name:@"currentUserUpdate" object:nil];
```

Ogni volta che da altre classi verrà modificata l’istanza di *User* (ad esempio al termine della registrazione, o dalla sezione del profilo, o a seguito dell’aggiornamento dei punti fedeltà dal backend), verrà anche lanciata una notifica con il nome *currentUserUpdate*: in questo modo, l’observer dichiarato farà eseguire all’*AppDelegate* il metodo definito nel selector, che si occupa di salvare l’istanza di *User* modificata sempre in *NSUserDefaults*.

Infine, la gestione delle notifiche push e delle notifiche locali. L’*AppDelegate* è la classe preposta per questo ruolo: viene gestita la richiesta dei permessi dall’utente e la registrazione al servizio di notifiche push, registrando il token

univoco collegato al dispositivo nella piattaforma di MobileBridge attraverso il rispettivo metodo della libreria. Ovviamente, per le notifiche locali non è richiesta una registrazione ad un servizio, in quanto vengono “lanciate” dal codice all’interno dell’app. La ricezione delle notifiche push e delle notifiche locali prevede due scenari differenti. Nel caso l’applicazione sia nello stato attivo (è in esecuzione e attiva su schermo), i segnali della notifica, quali banner e suono, non verrebbero eseguiti: in questo caso, quindi, per avvisare l’utente viene fatto il catch della notifica e visualizzato all’interno dell’applicazione una alert view con il contenuto della notifica stessa. Se l’utente decide di vedere i dettagli della notifica, al click sul pulsante dell>alert view viene postata una notifica all’interno del notification center, con oggetto il dictionary contenente i dati della notifica push: in questo modo l’observer, dichiarato in un view controller differente come spiegato più avanti, cattura la notifica, eseguendo il metodo predisposto. Nel caso in cui, invece, l’applicazione non sia nello stato attivo, ma viene aperta o rimessa in attivo a seguito di una iterazione dell’utente con la notifica push, viene direttamente postata una notifica all’interno del notification center seguendo il processo appena descritto.

4.3 View e Controller

Quattro controllers utilizzati in varie parti dell’applicazioni sono: PromotionViewController, SurveyViewController, PollViewController, RichMessageViewController, tutte sottoclassi di UIViewController. Il loro funzionamento è sostanzialmente simile: utilizzando le classi model precedentemente descritte, ognuno dichiara una property rispettivamente di tipo Promotion, Survey, Poll e RichMessage. Sfruttando gli attributi “ID” e “type”, i controller inizializzano le rispettive views fornite dal framework MB, ossia MB_PromotionView, MB_PollView, MB_SurveyView e MB_RichMessageView: è quindi il framework che, avendo l’id e il type, gestisce il caricamento e contenuto delle quattro views. I controllers implementano infine, oltre ad operazioni di minore importanza, i metodi dei protocolli MB_PromotionViewDelegate, MB_SurveyViewDelegate, MB_PollViewDelegate e MB_RichMessageViewDelegate, che si occupano di garantire il funzionamento delle views.

Proseguendo nella descrizione del codice, il TabBarController, il controller iniziale dell’applicazione, è definito e gestito direttamente all’interno dello Storyboard del progetto. Sono quattro i tab all’interno dei quali la navigazione fra diversi controller è gestita dall’UINavigationController standard.

HomeController, sottoclasse di UIViewController, è il “root view controller” del UINavigationController collegato al primo tab. Gestisce due views all’interno della sua view principale: BannerView e HomeView. BannerView, utilizzando il costruttore fornito dal framework MB, inizializza la view del banner, l’oggetto che fa scorrere le immagini di promozioni o sondaggi impostate nel CMS; l’impostazioni di immagini per il banner è opzionale, viene attivata per quelle promozioni a cui si vuole dare risalto nella home. HomeView, invece, gestisce la view, e i suoi elementi grafici, posizionata nella metà inferiore della view principale. In particolare: label e button relativi alla lista completa delle promozioni pubbliche; label, button e image del riepilogo del profilo utente; e l’immagine di sfondo a cui viene applicato un effetto di blurring. HomeController gestisce il touch su un banner e l’aggiornamento dei punti fedeltà dell’utente implementando i metodi dei due delegates MB_BannerView-Delegate e MB_PointsDataDelegate forniti dal framework. Oltre a ciò, il controller della home adempie ad altri due compiti importanti: carica, controllando un valore booleano memorizzato nell’oggetto NSUserDefaults, il WelcomeViewController, carica il PromotionsViewController al click sul relativo UIButton e soprattutto gestisce il caricamento delle views e dei controllers corretti all’arrivo di una notifica push o locale, attraverso la dichiarazione di un observer del notification center che fa il “catch” della notifica lanciata dall’AppDelegate, eseguendo il metodo predisposto a leggere i contenuti del dictionary della notifica che a sua volta inizializza il model e il controller appropriato.

WelcomeViewController è il controller presentato dalla home al primo avvio dell’applicazione. Gestisce tre funzionalità: l’inserimento dei dati dell’utente, implementando i metodi dell’UITextFieldDelegate, l’aggiornamento con i dati inseriti dell’istanza di User, lanciando inoltre una notifica nel notification center per richiederne il salvataggio, e l’invio dei dati alla piattaforma MobileBridge utilizzando i metodi forniti dalla libreria (vengono inviati data di nascita e sesso dell’utente, oltre al settaggio di alcune impostazioni) attraverso l’handler definito nell’AppDelegate. Il pulsante per passare alla schermata successiva viene animato e soprattutto attivato solo se tutti e 4 i campi richiesti sono stati compilati; la view è gestita interamente attraverso l’interface builder. Al touch sul button, viene presentata la view successiva, controllata dal WelcomeDetailViewController responsabile del funzionamento di due azioni alternative: la dismissione della view oppure la condivisione dell’applicazione. La condivisione viene gestita attraverso l’UIActivityViewController dell’SDK di iOS: in caso di successo, l’evento viene tracciato dal framework attraverso un metodo richiamato dall’handler della libreria.

PromotionsViewController, sottoclasse di UITableViewController, si occupa di caricare e visualizzare la lista delle promozioni pubbliche, e viene richiamato dalla home. La richiesta dei dati avviene attraverso l'handler della libreria MB definito nell'AppDelegate, in questo modo:

```
[appDelegate.mbLib MB_GetPublicPromotionData:self];
```

I dati della risposta vengono gestiti dal PromotionsViewController in quanto viene definito come "MB_GetPublicPromotionDataDelegate", attraverso il relativo metodo da implementare:

```
- (void)MB_GetPublicPromotionDataFinished:(NSArray*)data
```

I dati ottenuti vengono modellati in tanti oggetti "Promotion" quanti sono gli elementi dell'array di risposta. Questa procedura è sempre uguale, o molto simile, ogni volta che, da un controller, viene richiamato un metodo attraverso l'handler della libreria definito nell'AppDelegate: nel prosieguo della descrizione non verrà quindi illustrata nuovamente. Proseguendo, i dati degli oggetti "Promotion" ottenuti vengono visualizzati sfruttando i metodi standard della UITableView: il controller fa quindi da tramite tra il model, istanza della classe Promotion, e la view, in questo caso PromotionTableViewCell. Alla selezione di una riga, viene istanziato un PromotionViewController, al quale viene passato l'oggetto Promotion contenuto nell'array di indice uguale all'indice della riga selezionata; è inoltre implementato l'aggiornamento della tabella attraverso un UIRefreshControl.

LocationsViewController, sottoclasse di UITableViewController, è il "root view controller" del UINavigationController collegato al secondo tab e si occupa del caricamento della lista dei negozi configurati nel CMS di MobileBridge. Viene quindi utilizzato l'handler della libreria per richiamare il metodo pubblico del framework da utilizzare, seguendo la procedura descritta per la lista delle promozioni pubbliche. Al touch su una riga, LocationsDetailViewController si occupa di visualizzare la view con i dettagli del negozio selezionato, che al contrario della view della promozione non è gestita direttamente dalla libreria. La view è composta da due sotto-view: un'istanza della classe MKMapView (fornita dal framework MapKit) che visualizza la mappa con la posizione geografica del negozio, conosciuta grazie alle informazioni contenute nell'istanza della classe MB_LocationType descritta in precedenza. E una istanza della classe LocationInfoView che visualizza una label con l'indirizzo del negozio e quattro UIButton collegati alle quattro azioni gestite dal controller: l'invio di una mail, utilizzando i metodi definiti dal protocollo del MFMailComposeViewController-Delegate, l'apertura dell'applicazione "Mappe" con la navigazione impostata

dalla posizione attuale dell'utente all'indirizzo del negozio, l'apertura del sito web o la chiamata al numero del negozio utilizzando gli "URL schema" supportati dall'UIKit.

WalletViewController, sottoclasse di UITableViewController, è il "root view controller" del UINavigationController collegato al terzo tab, responsabile della visualizzazione della lista delle promozioni salvate nel wallet dall'utente. Nella navigation bar del controller è istanziato un UISegmentedControl: viene utilizzato l'handler della libreria MB per richiamare il metodo pubblico del framework che si occupa di scaricare gli elementi del wallet in base al "segmento" selezionato dall'utente, in base cioè alla tipologia della promozione che può essere attiva, usata o scaduta. Al touch su di una riga della table view viene istanziato un PromotionViewController. Utilizzando un'istanza della classe UILongPressGestureRecognizer invece, al touch prolungato (di un secondo) vengono visualizzate informazioni aggiuntive sulla promozione selezionata in una alert view; è gestito infine l'aggiornamento dei dati visualizzati nella tabella attraverso un UIRefreshControl.

ProfileViewController, sottoclasse di UIViewController, è il "root view controller" del UINavigationController collegato al quarto tab. Gestisce quattro operazioni principali: la visualizzazione dell'InboxView, la visualizzazione dei dati del profilo utente, la visualizzazione con possibilità di aggiornamento dei punti fedeltà, la gestione delle preferenze dell'applicazione legate alla piattaforma MB. Per quanto riguarda l'InboxView, è una delle cinque views gestite dal framework: l'inizializzazione dell'oggetto avviene quindi utilizzando il costruttore fornito dalla libreria, collegata al touch su un bar button item. Il controller è definito come MB_InboxViewDelegate: viene gestito, attraverso il relativo metodo implementato, il touch su un elemento dell'inbox, inizializzando il controller appropriato fra i quattro descritti inizialmente. Per le altre tre funzionalità, il controller viene definito come UITableViewDelegate e UITableViewDataSource della table view inserita all'interno della view principale, con tre diverse UITableViewCell. L'aggiornamento dei punti fedeltà e delle impostazioni dell'utente viene gestito utilizzando i metodi predisposti dal framework attraverso l'handler della libreria. Oltre alla visualizzazione dei dati del profilo, è possibile la modifica: al touch sul button predisposto viene istanziato il ProfileEditViewController, che gestisce la view successiva. E' un controller sostanzialmente simile al WelcomeViewController già descritto: gestisce la modifica dei dati dell'utente, permettendone il salvataggio solo se tutti i campi sono compilati.

Altri controllers di minore importanza sono l'InfoViewController, responsabile della visualizzazione della view contenente le informazioni dell'app e richiamato attraverso un button nella navigation bar della home, e lo StoreViewController, utile per gestire la view caricata all'arrivo di una notifica locale inviata in prossimità dell'iBeacon impostato nell'AppDelegate.

Infine, per quanto riguarda le views sviluppate di minore importanza, CustomAlertView è l>alert view sottoclasse di UIAlertView utilizzata per trasmettere, all'arrivo di una notifica con l'app attiva, il NSDictionary contenente i dati della notifica stessa dall'AppDelegate al HomeController attraverso i seguenti due metodi pubblici:

- *(void)setPayload:(NSDictionary *)pPayload;*
- *(NSDictionary *)getPayload;*

Ci sono poi le views, tutte sottoclassi di UITableViewCell, che definiscono gli elementi grafici per le righe delle varie table view, il cui layout viene gestito anche attraverso l'interface builder: LocationTableViewCell, LoyaltyTableViewCell, ProfileTableViewCell, PromotionTableViewCell, SettingTableViewCell, WalletTableViewCell.

4.4 Scelte implementative

Dopo aver descritto in maniera generale tutto il codice sviluppato, in questo paragrafo vengono spiegate le scelte implementative più importanti e non ancora discusse in precedenza, assunte durante la realizzazione di Promotions.

Tutto lo sviluppo è stato ovviamente finalizzato al raggiungimento degli obiettivi definiti durante la fase di progettazione. L'utilizzo della piattaforma MobileBridge è stato concordato all'inizio del tirocinio, con il tutor aziendale, in quanto le due aziende, GetConnected e MobileBridge, avevano in atto una collaborazione che mi ha permesso di utilizzare la versione beta della piattaforma; non sono state prese in considerazione ipotesi alternative. La gestione della libreria nel codice sviluppato ha seguito le linee guida tracciate nella documentazione.

Si è deciso di utilizzare la tecnologia iBeacon come tecnologia di prossimità per i casi d'uso interessati, e non altre soluzioni, in un primo momento perché era la tecnologia gestita all'interno della piattaforma; quando si è riscontrato il non funzionamento, ho comunque sviluppato questa integrazione perché mi è sembrata più interessante e soprattutto meglio documentata e integrata all'interno dell'SDK iOS, rispetto ad altre soluzioni che avrebbero potuto portarmi

a raggiungere gli stessi obiettivi. Da un punto di vista pratico inoltre, rispetto ad esempio alla tecnologia NFC, i beacons assicurano un raggio di azione notevolmente più ampio, oltre ad una durata dell'autonomia molto prolungata. Certo, il grosso limite nell'uso quotidiano della tecnologia e di conseguenza nell'applicazione dei casi d'uso previsti è sicuramente la necessità di interagire con un iPhone con il bluetooth attivo; così come potenzialmente limitati sono quei casi d'uso che richiedono l'iterazione con la posizione dell'utente, anche con l'app non in esecuzione, per determinate zone geografiche. In generale, comunque, l'applicazione e il customer journey prevedono una tale combinazione di casistiche (eventi e funzionalità interne all'app, geofencing tramite iBeacon o coordinate geografiche, notifiche push) da poter sopperire ad eventuali limitazioni causate dalle scelte dell'utente.

La piattaforma di mobile marketing automation e la tecnologia di prossimità hanno rappresentato certamente le due scelte fondamentali per il progetto. Altre decisioni minori, riguardanti lo sviluppo del codice, sono state:

- l'utilizzo di UserDefaults per il salvataggio locale dei dati sul profilo utente: è stata scartata l'ipotesi di implementazione attraverso un database vista la quantità limitata di dati da salvare;
- la gestione delle notifiche push è stata divisa tra AppDelegate e HomeViewController per lasciare a quest'ultimo i compiti inerenti l'inizializzazione dei controller e delle view in base alle informazioni ricevute;
- il notification center è stato utilizzato come strumento di notifica di eventi fra diversi controller, oltre ai delegates definiti dal framework MB, data la facilità di utilizzo e la conoscenza pregressa dell'argomento.

Capitolo 5

Promotions: simulazioni di utilizzo

Una volta completato lo sviluppo dell'applicazione, sono passato a provarla nella sua completezza, all'interno del customer journey ideato inizialmente e quindi applicando i diversi casi d'uso in esso contenuti.

L'applicazione, così come è stata sviluppata, potrebbe essere adatta per l'utilizzo in molteplici contesti o target, non avendo personalizzazioni grafiche o funzionali tali da escludere certe aziende o brand. E' stato deciso, come descritto nei capitoli dell'introduzione e della progettazione, di avere come contesto per la simulazione un brand che opera nel settore dell'alta moda, in generale dell'abbigliamento, e in particolare i contenuti della simulazione si riferiscono quasi totalmente ad un brand con il quale l'azienda in cui ho svolto il tirocinio ha una partnership. Per quei casi d'uso che richiedevano la definizione di un contesto geografico per il negozio, è stato utilizzato l'indirizzo presso cui ho svolto il tirocinio, che quindi simula il negozio stesso, utilizzando inoltre un beacon messi a disposizione dall'azienda per simulare l'entrata del negozio. L'attivazione delle promozioni non è avvenuta tramite la lettura del barcode, non disponendo di un sistema collegato, ma è stata anch'essa simulata tramite l'inserimento di un codice all'interno dell>alert.

5.1 Impostazione del CMS

Per prima cosa, sono state settate sul CMS di Mobile Bridge tutte le impostazioni e i contenuti utili alla simulazione. In particolare:

- alcune promozioni erano già presenti, sono state aggiunte quelle per il benvenuto, per il raggiungimento della soglia punti, per i passaggi ripetuti. Per ognuna è stato impostato il titolo, la descrizione, la tipologia (sconto), lo sconto in percentuale, la durata (standard, di 6 mesi), le attivazioni disponibili per utente (una), la modalità di attivazione (inserimento del codice o scansione codice a barre) e il contenuto tramite l'interface builder. Per la promozione fedeltà è stata anche impostata la quantità di punti da sottrarre all'utente al momento dell'attivazione. Le promozioni già presenti erano definite come pubbliche, quindi al contrario di quelle da me inserite comparivano nella lista delle promozioni consultabili dall'app, e due di esse avevano attiva l'impostazione per l'auto-salvataggio nel wallet di tutti gli utenti;
- sono stati creati con titolo, descrizione e contenuto i messaggi per i punti e la promozione di benvenuto e il sondaggio da sottoporre all'uscita dal negozio; è stato aggiunto un negozio, come descritto in precedenza, oltre ai due che erano già presenti;
- sono state create le automation e i profili utili ai casi d'uso. Ho creato l'automation per i punti fedeltà di benvenuto, che ha come trigger la prima apertura dell'app, come oggetto il messaggio creato in precedenza e l'assegnazione di 200 punti; l'automation per la promozione di benvenuto, che ha come trigger l'avvenuta condivisione dell'app (come spiegato nel capitolo 5 sullo sviluppo) e come oggetto la promozione creata; l'automation per la promozione fedeltà, che ha come trigger il raggiungimento di 1000 punti e come oggetto la promozione precedentemente creata; l'automation per il sondaggio, che ha come trigger l'uscita da un negozio, come oggetto il sondaggio e attivabile 1 volta sola per utente. Un particolare profilo, cioè gli utenti di 27 anni, è stato utilizzato per targettizzare l'invio tramite notifica push di una promozione speciale, per simulare ad esempio promozioni inviate in periodo di saldi o all'inizio di nuove stagioni. Per la situazione riguardante i passaggi ripetuti, non è stato possibile creare un'automation per la mancanza del trigger adeguato. Ho quindi creato un profilo target che selezionasse gli utenti che non fossero entrati negli ultimi sei mesi in nessuna delle "locations" definite, ma che fossero stati presenti almeno una volta in un'area con un raggio di 100 metri dalla location definita a Bologna. Successivamente ho

creato la notifica push selezionando il profilo creato e una promozione disponibile, abilitando la ricorsione della notifica ogni mese;

- nella sezione “Loyalty Points” è stato possibile definire le regole per l’assegnazione di punti fedeltà, a parte ovviamente i punti di benvenuto gestiti dall’automation. Ogni regola è caratterizzata da un evento, selezionabile da una lista di eventi gestiti dalla piattaforma, una quantità di punti, la quantità massima di punti ottenibile con quella regola e l’intervallo di tempo minimo in cui è concessa la ripetizione. Ho definite le regole per: l’entrata nel raggio di un beacon, quindi l’entrata in un negozio (20 punti, intervallo di 7 giorni), l’apertura dell’applicazione (5 punti, intervallo di 7 giorni), il salvataggio di una promozione (10 punti, intervallo di 3 giorni), l’attivazione di una promozione (15 punti, intervallo di 3 giorni), la risposta ad un sondaggio (15 punti, intervallo di un mese), la visualizzazione di un messaggio (5 punti, intervallo di 14 giorni).

5.2 Simulazioni di utilizzo

La prima simulazione è iniziata con l’installazione dell’app su un iPhone 5, con il bluetooth attivo: al primo avvio, come ci si aspettava, è apparsa la schermata dove poter inserire i dati del proprio profilo. Dopo aver dato i permessi per ricevere le notifiche push e abilitare i servizi di localizzazione richiesti, è arrivata la notifica sotto forma di avviso, essendo l’applicazione aperta, che informava dell’accredito dei punti fedeltà bonus di benvenuto (Figure 5.1 - 5.2).

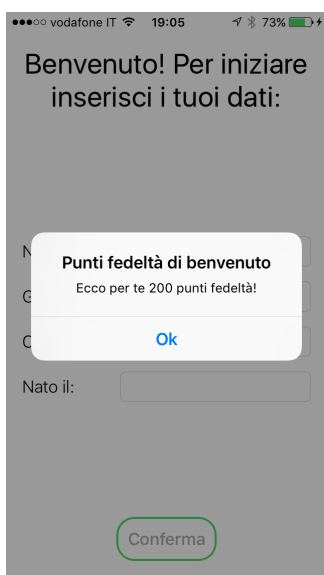


Figura 5.1: Schermata e notifica di benvenuto

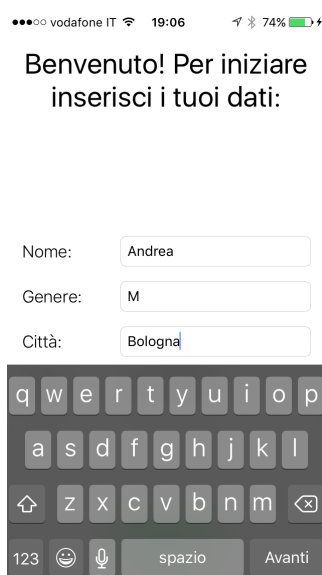


Figura 5.2: Inserimento dei dati per il profilo



Figura 5.3: Condivisione dell'app

Una volta completato il profilo ed essere passato alla schermata successiva, si è deciso di effettuare la condivisione richiesta tramite l'invio di una mail; a questo punto la schermata di benvenuto ha lasciato spazio alla home dell'applicazione ed è arrivata la seconda notifica, contenente la promozione speciale di benvenuto che è stata visualizzata e salvata nel wallet, vedi le Figure 5.4 - 5.6. Con un'altra installazione si è fatta la contro-prova e, come ci si aspettava, non effettuando la condivisione la notifica con la promozione non è arrivata.

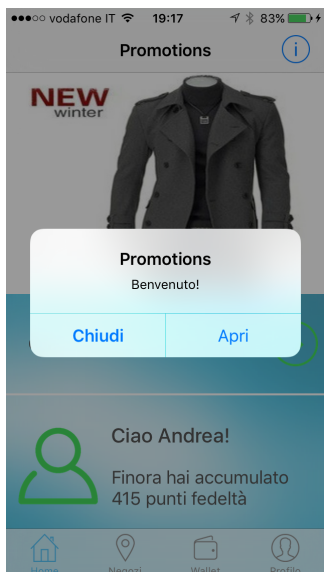


Figura 5.4: Notifica con promozione di benvenuto



Figura 5.5: La schermata con la promozione

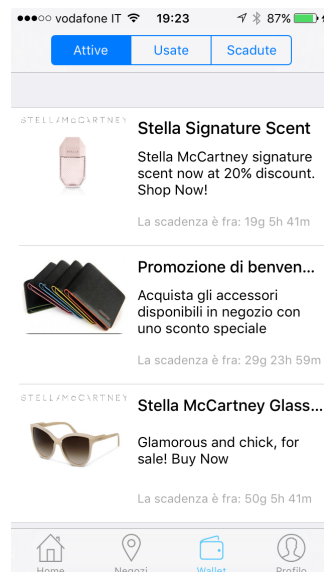


Figura 5.6: Wallet con le promozioni salvate

Si sono poi provate le varie funzionalità di base presenti nelle 4 sezioni dell'app, di cui si riportano tre screenshots di esempio nelle Figure 5.7, 5.8 e 5.9.

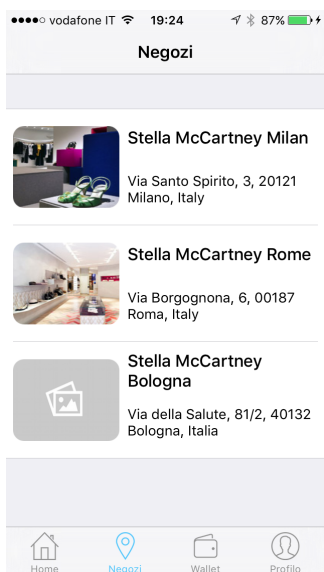


Figura 5.7: I negozi della catena

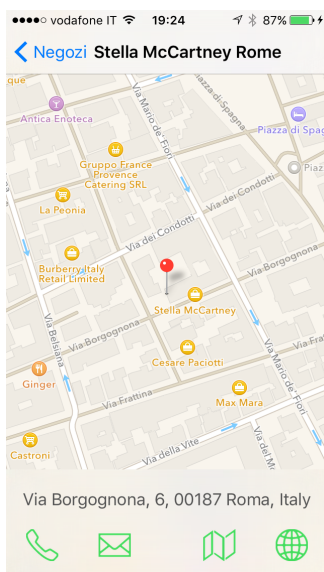


Figura 5.8: Dettaglio del negozio selezionato

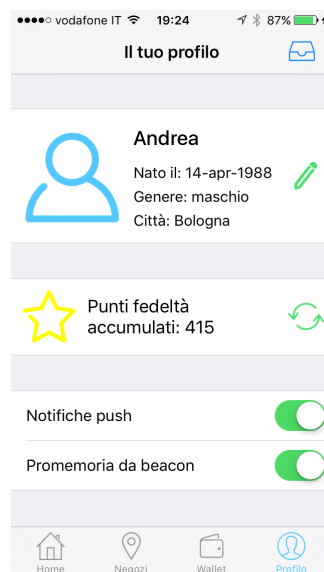


Figura 5.9: La schermata del profilo utente

Trovandosi inizialmente all'interno del recinto geografico caratterizzante uno store, una volta al di fuori del "recinto" è arrivata la notifica che chiedeva di rispondere ad un sondaggio sull'esperienza nel negozio, sondaggio a cui ho risposto (Figure 5.10 e 5.11). Con un'altra simulazione, sempre sullo stesso dispositivo ma con un'altra installazione, è stata testata la situazione dei passaggi ripetuti: come scritto nel precedente paragrafo, era richiesto un solo passaggio nell'area stabilita, e la notifica push è arrivata correttamente, come si può vedere nella Figura 5.12.

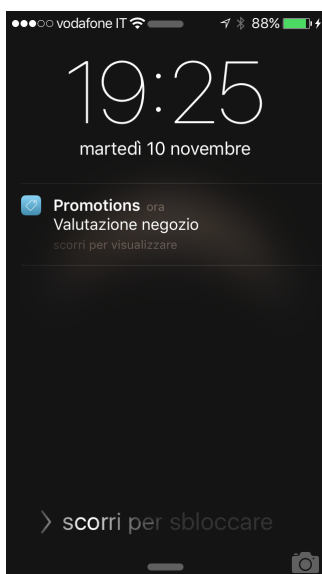


Figura 5.10: Notifica all'uscita del negozio

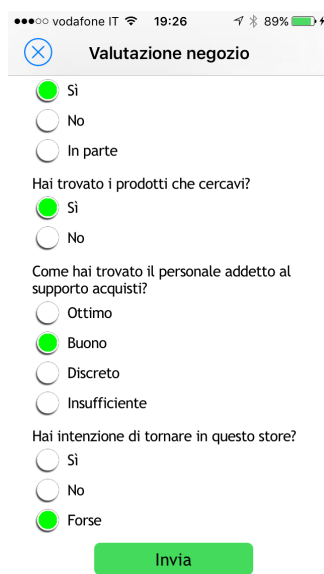


Figura 5.11: Il sondaggio inviato con la notifica

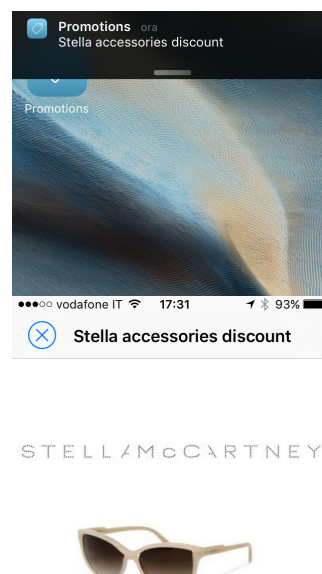


Figura 5.12: Promozione per passaggi ripetuti

Un'altra prova è stata effettuata inviando una notifica contenente una promozione ad un target preciso (in questo caso, l'età di 27 anni) dal CMS: la notifica e il conseguente comportamento dell'applicazione hanno funzionato come previsto, non ne vengono riportate le schermate perché la promozione inviata è la stessa di quella utilizzata nel caso d'uso dei passaggi ripetuti.

Successivamente, in una diversa simulazione, si sono aumentati i punti fedeltà legati al profilo creato, utilizzando il CMS di Mobile Bridge, in modo da arrivare alla quota di 1000 punti, utile per attivare la promozione fedeltà, arrivata sempre tramite una notifica push: una volta utilizzata attraverso l'inserimento di un codice prestabilito, i punti sono stati scalati dal profilo (vedi Figure 5.13 - 5.15).

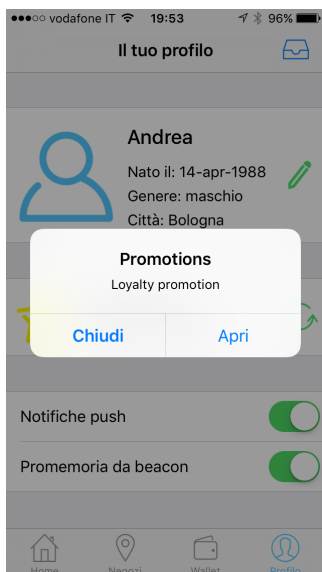


Figura 5.13: Notifica con promozione fedeltà



Figura 5.14: Il contenuto della promozione

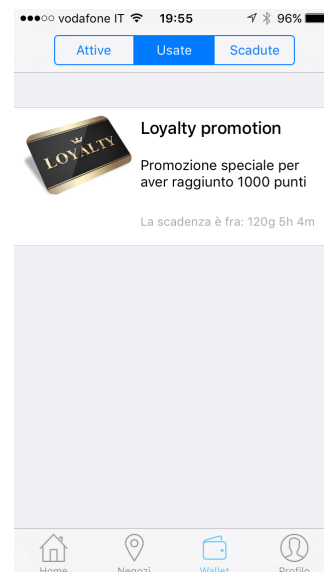


Figura 5.15: Utilizzo della promozione fedeltà

Per quanto riguarda i casi d'uso che prevedevano l'iterazione con un beacon gestiti direttamente all'interno dell'applicazione, sono state effettuate due prove: nella prima, una volta arrivato in prossimità del beacon che rappresentava l'ipotetica entrata del negozio, è arrivata correttamente la notifica locale con la presentazione del negozio stesso (Figure 5.16 e 5.17), mentre nel secondo caso, ad un nuovo arrivo nel recinto geografico determinato dal beacon, è arrivata, come previsto, la notifica push contenente la promozione di benvenuto inviata anche all'avvenuta condivisione, successivamente alla creazione del profilo (Figura 5.18).



Figura 5.16: Notifica in prossimità del negozio



Figura 5.17: La view di presentazione

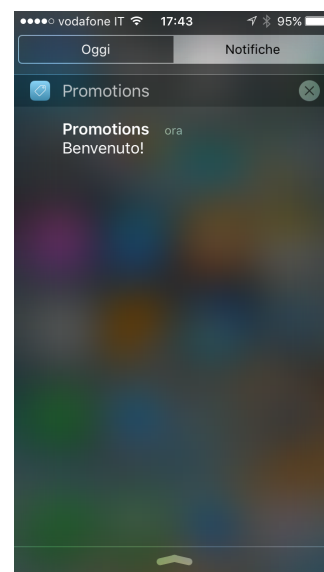


Figura 5.18: Notifica per promozione benvenuto

L'unica situazione prevista che non è stato possibile testare è stata quella relativa ai reminders delle promozioni in base alla prossimità di un beacon, a causa del già citato problema nella gestione dei beacons all'interno della piattaforma MB: va detto che la funzionalità potrebbe essere replicata, sempre all'interno della piattaforma, con i reminders legati all'entrata di una locations, sfruttando cioè la geo-localizzazione in maniera simile a quanto fatto per il sondaggio.

5.3 Presentazione dell'applicazione

Oltre alle simulazioni di utilizzo descritte, il customer journey e l'applicazione sviluppata sono stati presentati in occasione dell'evento "Dal Fashion all'e-Fashion all'Omni-Channel Fashion" tenutosi a Milano il 5 novembre 2015 e organizzato dall'azienda Hybris software, ottenendo commenti e pareri positivi.

Il tema dell'evento era: "Una riflessione sul nuovo concetto di Fashion & Luxury alla luce dell'evoluzione del consumatore e delle moderne tecnologie a supporto del posizionamento, della vendita e della fedeltà verso il brand. Come deve evolvere la proposta e la vendita dei beni di lusso nell'era 3.0? Come sarà il negozio del futuro? Come creare e mantenere un legame emozionale con i consumatori e trasformarli in brand ambassador? Queste le domande cui vogliamo dare una risposta, discutendo di come la tecnologia rappresenti un'opportunità enorme per fidelizzare un cliente sempre più digitale e attento a valori quali esclusività, personalizzazione dell'esperienza e qualità. Un contesto in cui, se l'online gioca un ruolo primario, lo store ha l'esigenza di rinnovare la propria funzione e il proprio ruolo nell'ambito di una sfida dove e-commerce, social network e mobility sono protagonisti." All'evento hanno partecipato rappresentanti dei maggiori brand della moda italiana oltre ad aziende e realtà operanti nel mondo dell'IT ed e-Commerce quali ad esempio netcomm (Consorzio del commercio elettronico italiano), contactlab e MobileBridge.

L'applicazione è stata presentata da Alessandro Rizzoli, co-relatore di questa tesi, tramite screenshots e descrizione in relazione al customer journey; una demo non è stata possibile dato che l'app non era ancora terminata e testata completamente. La presentazione è avvenuta in quanto il progetto e le finalità di questa tesi erano particolarmente aderenti ai temi trattati durante l'evento, come dimostrazione delle possibilità aperte per le aziende dal mondo mobile in relazione a clienti sempre più digitali.

Capitolo 6

Conclusioni

Vengono proposte le conclusioni del presente elaborato, composte da un elenco di possibili sviluppi futuri, miglioramenti e da una serie di considerazioni finali.

6.1 Possibili sviluppi

L'applicazione sviluppata rispetta i requisiti funzionali previsti, ma potrebbe essere integrata con ulteriori funzionalità, in modo da renderla ancor più completa. Ho immaginato quindi alcuni possibili sviluppi del progetto:

- informazioni sul catalogo: si potrebbe integrare una sezione dell'app in cui poter visualizzare i principali prodotti offerti dal negozio (o dalla catena di negozi) per offrire all'utente già all'interno dell'applicazione una panoramica sull'offerta e, ad esempio, sui nuovi arrivi;
- piattaforma MobileBridge: alcune funzionalità, come i video e i tag per le promozioni, non sono state integrate perché ritenute non indispensabili ai fini del progetto, ma potrebbero completare l'esperienza di utilizzo dell'app Promotions e l'integrazione con la piattaforma;
- integrazione con un e-commerce: integrare, all'interno dell'app, una piattaforma di e-commerce legata allo stesso negozio porterebbe, oltre ad

offrire un'esperienza unificata e completa al cliente, ad aggiungere ulteriori casi d'uso e possibilità di marketing multicanale. Ad esempio, all'arrivo del cliente in un negozio (quindi sfruttando la geo-localizzazione o la prossimità ad un beacon) si potrebbero proporre promozioni o messaggi targettizzati in base alla lista dei desideri dell'account del cliente utilizzato nell'e-commerce. Questa possibilità era stata prevista nelle fasi iniziali del tirocinio durante il quale ho sviluppato questo progetto, ma le difficoltà di implementazione, in relazione anche al tempo a disposizione, ne hanno impedito lo sviluppo;

- integrazione con ulteriori funzionalità di iOS: per migliorare la user experience, potrebbero essere integrate ulteriori funzionalità native offerte dall'SDK di iOS, quali ad esempio l'integrazione con la ricerca Spotlight per i contenuti salvati nell'applicazione (promozioni, sondaggi, negozi), lo sviluppo di un widget per il Centro Notifiche che visualizzi le promozioni salvate in scadenza, il supporto alle gestures del 3D Touch integrato negli iPhone di ultima generazione;
- versione Android: la piattaforma di Mobile Bridge offre un SDK anche per il sistema operativo Android, quindi potrebbe essere utile svilupparne un'applicazione equivalente, in modo da ampliare in maniera notevole i potenziali utilizzatori e quindi aumentare l'impatto delle politiche di marketing previste nel customer journey.

6.2 Considerazioni finali

In conclusione, l'utilizzo di smartphone e di tecnologie ad esso collegato per proporre nuove soluzioni di marketing, migliorando e arricchendo la relazione instaurata con il cliente e aumentando le possibilità di engagement, è agli inizi, soprattutto in Italia.

L'obiettivo iniziale della tesi, cioè sviluppare un'applicazione iOS per un ipotetico brand, che implementasse politiche innovative di marketing di prossimità e automation previste da un determinato customer journey, si può dire che è stato raggiunto, attraverso un lavoro iniziato dall'ideazione e progettazione, proseguito con lo sviluppo e infine con le simulazioni di utilizzo.

Seppur Promotions sia integrata con una piattaforma attualmente in beta, non potendone sfruttare appieno le funzionalità, situazione che ha però portato all'approfondimento della tecnologia iBeacon e allo sviluppo autonomo di alcune soluzioni, e seppur l'applicazione possa essere migliorata in varie parti come illustrato nel paragrafo precedente, il risultato finale raggiunto può quindi considerarsi positivo, come dimostrano le simulazioni di utilizzo effettuate e la

presentazione avvenuta a Milano in presenza di importanti marchi internazionali. Promotions offre appunto un esempio completo ed un possibile punto di partenza futuro per politiche di mobile marketing automation e di prossimità previste da un determinato customer journey che sfruttino i vantaggi, le potenzialità e le tecnologie di un'applicazione nativa per smartphone.

Bibliografia e Sitografia

- [1] Cindy Krum, *“Mobile Marketing: trovate i vostri clienti dovunque siano”*, Hoepli, 2011
- [2] Techcrunch, *“Majority Of Digital Media Consumption Now Takes Place In Mobile Apps”*, <http://techcrunch.com/2014/08/21/majority-of-digital-media-consumption-now-takes-place-in-mobile-apps>, 21 agosto 2014
- [3] Alessio Semoli, *“Marketing automation: Guida completa per automatizzare il tuo business online”*, Hoepli, 2015
- [4] *“Mobile marketing automation”*, https://en.wikipedia.org/wiki/Mobile_marketing_automation
- [5] VentureBeat, *“Mobile marketing automation: 1.5% penetration now, could triple in 2015”*, <http://venturebeat.com/2015/01/15/mobile-marketing-automation-1-5-penetration-now-could-triple-in-2015>, 15 gennaio 2015
- [6] *“Marketing di prossimità”*, https://it.wikipedia.org/wiki/Marketing_di_prossimità
- [7] *“Geo-fence”*, <https://en.wikipedia.org/wiki/Geo-fence>
- [8] Wired, *“CheckBonus e Open Service, lo shopping in negozio si fa con lo smartphone”*, <http://www.wired.it/mobile/smartphone/2014/06/13/proximity-marketing-e-ibeacon-il-futuro-dello-shopping-nei-negozi-fisici>, 13 giugno 2014
- [9] *“Kontakt.io Beacons”*, <https://support.kontakt.io/hc/en-gb/categories/200322471-Kontakt-io-Beacons>

- [10] “*Getting Started with iBeacon*”, <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>, 2 giugno 2014
- [11] “*User Stories: An Agile Introduction*”, <http://www.agilemodeling.com/artifacts/userStory.htm>
- [12] “*Core Location Framework Reference*”, https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/

Ringraziamenti

Il primo e più importante ringraziamento lo faccio alla mia famiglia, che mi ha sempre sostenuto e permesso di arrivare fino a qui. Un grazie grande anche ai parenti, amici e compagni di corso, che in maniera diversa ma sempre preziosa mi hanno accompagnato in questi anni.

Per la realizzazione di questa tesi, ringrazio il relatore professor Bononi per il supporto e il co-relatore Alessandro Rizzoli per l'opportunità di sviluppo che mi è stata offerta.