

UNIVERSITÀ DEGLI STUDI DI TRENTO

Facoltà di Scienze Matematiche, Fisiche e Naturali



UNIVERSITY OF TRENTO - Italy

Corso di Laurea in Informatica

Sincronizzazione di database centralizzati tramite dispositivi mobili Android: il caso di studio dell'applicazione What's Up?

Relatore: Maurizio Marchese
Correlatore: Cristhian Parra

Laureando: Maurizio Cainelli

Anno accademico 2011 - 2012

INDICE

1. Introduzione	3
1.1 Contesto della tesi	3
1.2 Lavoro della tesi	4
2. Stato dell'arte dei Social Network	5
2.1 Facebook (FB)	6
2.2 Google plus (G+)	7
2.3 Twitter	9
2.4 LinkedIn	10
2.5 Youtube	12
2.6 Applicazioni per i dispositivi mobili	13
3. Progetto di ricerca Life Participation	16
3.1 Applicazioni sviluppate	17
3.1.1 Active Lifestyle	18
3.1.2 Active Brain	19
3.1.3 Come Along	20
3.1.4 Liquid Galleries	22
3.1.5 What's Up?	24
4. Sincronizzazione di risorse tra dispositivi Android	27
4.1 Introduzione ad Android	27
4.2 Tecnologie utilizzate	28
4.2.1 MQTT (MQ Telemetry Transport)	29
4.2.2 C2DM (Cloud to Device Messaging)	30
4.2.3 GCM (Google Cloud Messaging)	33
4.2.4 Uniqush	36
4.3 Architettura	40
4.3.1 Use Case Diagram	41
4.3.2 Activity Diagram	43
4.4 Immagini dell'applicazione sviluppata	49
5. Risultati e conclusioni	54
5.1 Risultati	54
5.2 Conclusioni	57
Bibliografia	58

Capitolo 1

Introduzione

In questo primo capitolo parlerò del contesto della tesi, ovvero di cosa tratteranno i prossimi capitoli, ed il lavoro della tesi svolto presso il progetto di ricerca Life Participation.

1.1 Contesto della tesi

In questo sottocapitolo andrò ad illustrare di cosa trattano i prossimi capitoli raccontando in sintesi il loro contenuto.

Nel secondo capitolo, andrò a parlare dello stato dell'arte dei Social Network in quanto l'applicazione da me sviluppata è inclusa in questo mondo. Andrò ad analizzare i Social Network più famosi come Facebook, Twitter, Google+, etc, andando ad analizzare la loro storia, a che gruppo di persone è rivolto, i servizi offerti, gli aspetti positivi e negativi. Infine andrò ad analizzare le loro applicazioni create per i dispositivi mobili andando a catturare i problemi che un utente anziano può trovare usando queste applicazioni.

Nel terzo capitolo, andrò a parlare del progetto di ricerca Life Participation raccontando gli obiettivi principali del gruppo di ricerca, le motivazioni che hanno spinto a sviluppare determinate applicazioni e le applicazioni sviluppate.

Nel quarto capitolo illustrerò il lavoro che ho effettuato con il progetto di ricerca Life Participation, ovvero la sincronizzazione di risorse tra dispositivi Android. Prima di tutto effettuerò un'introduzione su Android, successivamente andrò ad analizzare ogni metodo di programmazione con il quale ho sviluppato l'applicazione partendo da MQTT fino ad arrivare all'ultimo metodo usato ovvero GCM e per ogni metodo di programmazione andrò ad analizzare i loro vantaggi e svantaggi. Di seguito analizzerò Uniquish, ovvero il server utilizzato per lo scambio di messaggi tra i dispositivi, mostrerò il protocollo utilizzato per il relativo scambio di messaggi ed infine farò vedere delle immagini relative all'applicazione.

Nel quinto capitolo, mostrerò i risultati ottenuti ovvero come mettere l'applicazione su *Google Play*, dirò a che *range* di utenti è indirizzata l'applicazione e trarrò delle conclusioni relative all'applicazione ovvero andrò a dire cosa le novità che si vogliono introdurre su quest'applicazione.

1.2 Lavoro della tesi

Il lavoro della tesi consiste nella spiegazione dell'applicazione sviluppata da me per il progetto di ricerca Life Participation ovvero un'applicazione che sia in grado di sincronizzare dei database centralizzati per poter sincronizzare le risorse tra vari dispositivi mobili Android.

Ho scelto di effettuare questo lavoro in quanto gli anziani si sono sempre più allontanati dal mondo tecnologico non per loro scelta, ma per colpa degli sviluppatori che creano sempre più siti e social network per avere un *range* di utenti giovane o comunque non troppo vecchio. Anche se attualmente il *range* di utenti attuale per i social network include pure gli utenti più anziani, le applicazioni per i dispositivi mobili non sono progettate con lo scopo di aiutare le persone più anziane ad avvicinarsi a questa nuova tecnologia che crea un grande *gap* tra gli utenti dei dispositivi mobili e le persone anziane.

Il progetto di ricerca Life Participation cerca sempre di diminuire questo *gap* tra le generazioni cercando di avvicinare gli utenti più anziani al mondo tecnologico creando applicazioni più facili e chiedendo le loro opinioni su ciò che vogliono e che non vogliono. Così facendo gli anziani si sentono parte di un gruppo e sono più invogliati a provare le applicazioni delle quali sono stati partecipi.

L'applicazione *Whats UP?* cerca di avvicinare queste due generazioni essendo una specie di social network ma molto semplificato. Il lavoro della mia tesi è stato quello di implementare un modo per sincronizzare le risorse tra i vari dispositivi, così facendo l'utente anziano può visualizzare le immagini in tempo reale con il proprio parente anche se non è presente con lui.

Capitolo 2

Stato dell'arte dei Social Network

In questo capitolo andremo ad approfondire lo stato attuale dei social network andando ad analizzare i più famosi, la loro struttura, cioè cosa offrono e cosa no, a che parte di utenti essi siano rivolti e la loro applicazione per i dispositivi mobili ma, prima di tutto questo, definiamo cosa s'intende con la parola social network (in italiano “rete sociale”).

Una rete sociale consiste in un qualsiasi gruppo di individui connessi tra loro da diversi legami sociali. Per gli esseri umani i legami vanno dalla conoscenza casuale, ai rapporti di lavoro, ai vincoli familiari.¹

I social network al momento più famosi, sono:

- Facebook;
- Google+;
- Twitter;
- LinkedIn;
- YouTube.

Facebook e Google+ hanno implementato il sistema di chat al loro interno mentre gli altri, no. La chat è un sistema che permette di effettuare una conversazione in tempo reale con un'altra persona, anche sconosciuta. Il luogo in cui venivano effettuate queste conversazioni viene definito, in termine tecnico, *chatroom*. La chat più famosa, per noi informatici, è IRC(*Internet Relay Chat*). Il sistema di chat si è poi sviluppato ed hanno introdotto nuove chat come *webchat* e i client di *istant messaging* come ICQ o MSN Messenger.²

Passiamo all'analisi più approfondita dei social network sopra citati.

¹ Cfr. http://it.wikipedia.org/wiki/Rete_sociale

² Cfr. <http://it.wikipedia.org/wiki/Chat>

2.1 Facebook (FB)

Facebook, nato nel 2004, è uno dei padri fondatori dei social network. Esso nacque da un'idea di Mark Zuckerberg il quale iniziò a programmare un sito web, sotto commissione, per l'università di Harvard. Questo sito web doveva essere un social network per i studenti di Harvard ma, Zuckerberg, volle ampliare questo sito web anche ai non studenti di Harvard e così nacque Facebook.³ All'inizio era adibito al sol scopo di conversare con una o più persone tramite l'invio di messaggi ad esse, ma poi, con il passare del tempo, si è evoluto, ha introdotto cose nuove come i *social games* e le videochiamate (effettuabili ad una sola persona alla volta). Effettuando queste modifiche, Zuckerberg, è andato a colpire un pubblico di utenti sempre più specifico che sono le persone medio-giovani, anche se, essendo sviluppato per giovani, l'utilizzo da parte di persone meno giovani (45-54 anni) è aumentato fino a raggiungere l'1/3 dei visitatori del canale mentre, gli utenti over 55, sono aumentati del 15%.⁴

Le migliorie apportate a Facebook, da quando è nato sono:

- agosto 2006: viene introdotta la funzione Note che permette agli utenti di pubblicare scritti e articoli in maniera simile a un blog. Sono attive le funzioni di tag e di inserimento immagini;
- maggio 2007: su Facebook è disponibile il Marketplace, che consente agli utenti di inserire gratuitamente annunci che risultano visibili solo da utenti presenti nella stessa rete;
- giugno 2009: introdotta la funzionalità username, che rende i profili accessibili tramite indirizzi semplificati;
- 21 aprile 2010: lanciato il tasto Mi piace, con il quale gli utenti possono esprimere apprezzamento su dei singoli contenuti;
- aprile 2011: la chat è stata arricchita da una funzione per effettuare chiamate vocali, che permette anche di lasciare messaggi in una segreteria vocale;
- 6 luglio 2011: lanciato il servizio di videochiamate che utilizza la tecnologia di Skype.

³ Cfr. <http://it.wikipedia.org/wiki/Facebook>

⁴ Cfr. <http://wearesocial.it/blog/2012/05/evoluzione-dei-social-media-italia-implicazioni-sulle-attivita-dei-brand/>

Le pagine Facebook sono scritte in linguaggio PHP e sono poi tradotte in C++ dal convertitore *HipHop for PHP*, ideato appositamente da Facebook. Grazie a questa tecnologia, Facebook dichiara un risparmio del 50% nell'utilizzo del processore dei propri server.

2.1.1 Riassumendo:

Anno di nascita	Stima utenti	Servizi offerti	Aspetti positivi	Aspetti negativi
2004	1000 mln al 4 ottobre 2012.	<ul style="list-style-type: none"> • Chat • Videochiamate • Messaggi • Note • Possibilità di far sapere che ti piace un post • Commentare i post dei tuoi amici • Possibilità di taggare i propri amici 	Essendo uno dei social network più diffusi, si può trovare facilmente qualsiasi persona con cui si vuole riprendere contatto. Si può chattare in tempo reale e non con le persone ed anche sapere le opinioni altrui tramite commenti e mi piace.	L'applicazione per mobile ha meno funzioni rispetto all'applicazione web (ad es: non puoi condividere link, non puoi chattare, non puoi avviare videochiamate).

2.2 Google plus (G+)

Google plus, nato nel 2011 e fondato da Google, ha reso più professionale un social network grazie a delle features innovative incrementando il suo range di utenti inglobando pure gli utenti che vogliono utilizzare un social network per lavoro, fare conferenze, etc. A differenza di altri social network che sono accessibili solo attraverso un unico sito web, Google descrive Google+ come un livello sociale (*social layer*) costituito non solo da un singolo sito, ma piuttosto da un “livello” generale (*overarching layer*) che copre molte delle sue proprietà online⁵.

Ciò che differenzia Google+ dagli altri social network sono le esclusive feature che può offrire come:

- le cerchie: esse sono dei gruppi nei quali includere determinate persone che fanno parte di quella “cerchia”;

⁵ Cfr. <http://en.wikipedia.org/wiki/Google%2B>

- gli *hangouts*: essi sono una delle *feature* che rende unico nel suo genere il social network di casa Google. Grazie a questa funzione Google+ offre la possibilità di effettuare videochiamate non solo ad una singola persona, come accade con Facebook, ma ad un gruppo di persone diventando, così, videoconferenze;
- la funzione *instant upload*: questa funzione riguarda il mondo dei device mobili. Essa permette di non perdere i video o le foto scattate sul telefonino, in quanto verranno inviate al proprio profilo Google+, salvate in un album privato, e, in seguito, si potrà decidere se rendere pubbliche queste immagini/video oppure no.

Ovviamente ci sono tante altre funzioni che Google+ può effettuare che sono molto simili ad altre già introdotte in questo mercato come i *social games*, *messenger*, *sparks*, etc.

Le tecnologie utilizzate, per sviluppare Google+, sono: Java servlet per la parte server, Javascript per l'interfaccia utente della parte browser e, per mantenere un bel URL nonostante Google+ sia un'applicazione AJAX, HTML5 History API.⁶

2.2.1 Riassumendo:

Anno di nascita	Stima utenti	Servizi offerti	Aspetti positivi	Aspetti negativi
2011	400 mln di utenti registrati a settembre 2012.	<ul style="list-style-type: none"> • Chat • Videochiamate • Videochiamate di gruppo • Messaggi • Note • Possibilità di far sapere che ti piace un post • Commentare i post dei tuoi amici • Possibilità di taggare i propri amici 	L'applicazione per mobile offre le stesse caratteristiche del sito web (sono supportate pure le videochiamate di gruppo).	<p>Per poter chattare bisogna utilizzare l'applicazione specifica ovvero Google Talk.</p> <p>Non è utilizzato come Facebook e questo è un peccato in quanto ha un alto potenziale (600 mln di persone di differenza).</p>

⁶ Cfr. <http://en.wikipedia.org/wiki/Google%2B#Technologies>

2.3 Twitter

Twitter è un servizio di social network e di microblogging che permette agli utenti di avere una pagina personale aggiornabile tramite testo con una lunghezza massima di 140 caratteri. Twitter è costruito totalmente su un'architettura Open Source (la API è aperta a tutti gli sviluppatori in modo da poter includere Twitter in applicazioni web, desktop o mobile). L'interfaccia di Twitter è scritta in *Ruby on Rails*, i messaggi vengono memorizzati in un software server programmato in *Scala*.⁷

La popolarità di Twitter ha visto una svolta con l'edizione del 2007 del South by Southwest festival: nei giorni dell'evento l'uso di Twitter è triplicato passando da 20.000 ad oltre 60.000 tweet al giorno. All'evento erano presenti due grandi schermi 60 pollici utilizzati esclusivamente per far scorrere lo stream dei tweet pubblici. Il festival fu un grande evento pubblicitario per il servizio e lo staff ricevette il Web Award prize.

Twitter modifica il concetto di amici introducendo i *followers*. I *followers* sono persone che vogliono sapere i tuoi aggiornamenti di stato, vedere le tue nuove foto e così via. A differenza della richiesta d'amicizia che ti chiede di accettare o meno una persona prima di condividere le proprie informazioni personali, colui che viene seguito, il *followed*, riceve solo una notifica che una determinata persona sta ricevendo i suoi aggiornamenti anche se una persona non la si conosce. Per togliere una determinata persona bisogna andare nella propria lista dei *follower*, cercarla, e poi bloccarla. Così facendo la persona bloccata non riceverà più nostre notizie e non potrà più interagire con noi.

Twitter viene usato da un'utenza medio-giovane. Essa è molto ampia in quanto su Twitter esiste poco spam la divulgazione di notizie è molto rapida grazie agli *Hashtags* che permettono di raggruppare determinati *tweet*, con lo stesso *hashtag*, e di visualizzarli tutti assieme.

⁷ Cfr. <http://it.wikipedia.org/wiki/Twitter>

2.3.1 Riassumendo:

Anno di nascita	Stima utenti	Servizi offerti	Aspetti positivi	Aspetti negativi
2006	500 mln di utenti registrati a febbraio 2012.	<ul style="list-style-type: none">• Messaggi• Commentare i post dei tuoi amici• Possibilità di taggare i propri amici• Possibilità di creare delle etichette da utilizzare nei propri <i>tweet</i> in modo da poter cercare una determinata categoria.	<p>L'applicazione per mobile offre le stesse caratteristiche del sito web.</p> <p>Gli hashtag sono molto utili in quanto si possono ricercare delle categorie specifiche identificate con #parola (nel caso del terremoto in Abruzzo, gli utenti Twitter hanno segnalato la notizia prima dei media tradizionali).</p>	<p>L'impossibilità di chattare, in tempo reale, è una grande pecca per questo social network.</p> <p>Il <i>tweet</i> è pubblicato solo se ha al massimo 140 caratteri.</p> <p>Non puoi far sapere al tuo <i>followed</i> che ti piace un suo post (come su FaceBook con i 'Mi piace').</p>

2.4 LinkedIn

LinkedIn è un social network per persone che cercano un'occupazione professionale, quindi il suo compito principale è quello di mettere in comunicazione un possibile candidato ed il proprio datore di lavoro.⁸

Una novità introdotta da LinkedIn è la possibilità di bloccare una persona, o limitarne l'account, se questa invia troppe richieste di collegamenti che vengono rifiutate (segnalata come *I don't know* oppure *spam*).

La lista di connessioni che si viene a formare tra contatti che hanno accettato i collegamenti, può essere usata in vari modi:

- la rete di un contatto è costruita dalle sue connessioni dirette, i quali possono avere altri collegamenti (*second-degree connections*) e, questi ultimi, possono avere altri collegamenti (*third-degree connections*).

⁸ Cfr. <http://en.wikipedia.org/wiki/LinkedIn>

Questo è un modo utile per poter introdurre qualcuno a persone che vogliono conoscere possibili candidati attraverso un contatto comune;

- gli utenti possono caricare il proprio Curriculum Vitae o creare il proprio profilo al fine di mostrare esperienze lavorative;
- può essere usata per trovare lavoro, persone o opportunità di *business* consigliato da qualcuno che è nella rete del contatto;
- i datori di lavoro possono indicare i lavori che cercano per i potenziali candidati.

Gli utenti che utilizzano LinkedIn come social network sono persone che cercano lavoro e lo vogliono trovare senza ricorrere alle agenzie di lavoro. L'utenza di LinkedIn è molto specifica in quanto non si può condividere qualcosa che ti piace, esprimere ciò che ti passa per la testa, come su FaceBook, Google+ o Twitter. Esso ha solamente la funzione di mettere in contatto, tra di loro, un disoccupato ed un possibile datore di lavoro.

2.4.1 Riassumendo:

Anno di nascita	Stima utenti	Servizi offerti	Aspetti positivi	Aspetti negativi
2002	175 mln di utenti registrati a giugno 2012 da 200 paesi differenti.	<ul style="list-style-type: none"> • Messaggi • Possibilità di ottenere un lavoro • Possibilità di caricare sul proprio profilo, il Curriculum Vitae in moda dare informazioni agli altri utenti • Possibilità di ricercare delle aziende che potrebbero essere interessate ad un determinato lavoro 	<p>Si può trovare lavoro attraverso i propri collegamenti (cosa che gli altri social network non fanno).</p> <p>E' per un pubblico specifico che cerca lavoro.</p>	<p>E' poco utilizzato dagli utenti che ricercano lavoro e preferiscono altri siti web.</p> <p>Non permette la chat istantanea tra possibile datore di lavoro e possibile lavoratore.</p>

2.5 YouTube

YouTube potrebbe essere l'unico della lista a non entrare nel gruppo di social network in quanto, la sua funzione principale, è quella di caricare video online e renderli visibili a tutte le persone che cercano un determinato video. YouTube entra nella lista dei social network, in quanto crea delle relazioni tra gli utenti di un determinato canale (*subscribers*) che guardano video, commentano e possono confrontarsi tra di loro. YouTube modifica il concetto di amici ed introduce il concetto di *subscribers*. I *subscribers* sono coloro che si sottoscrivono ad un canale per poter visualizzare in prima pagina i video dei canali in cui fanno parte. La persona che gestisce il canale, un po' come su Twitter, non può decidere se accettare o meno un *subscriber* e non può nemmeno rimuoverlo.

YouTube è utilizzato anche dalle più grandi *corporation* come CBS, BBC, VEVO, Hulu, e altre organizzazioni che offrono qualche materiale proprio. La tecnologia utilizzata per sviluppare YouTube è Adobe Flash Video e HTML5.⁹

Quasi nessun utente considera YouTube come un social network, ma viene considerato più come un sito dove condividere i propri video. YouTube è focalizzato ad una cerchia di utenti medio-giovani maggiori di 18 anni che passano il loro tempo facendo video.

2.5.1 Riassumendo:

Anno di nascita	Stima utenti	Servizi offerti	Aspetti positivi	Aspetti negativi
2005	100 mln di utenti registrati circa.	<ul style="list-style-type: none">• Messaggi• Possibilità di caricare video• Possibilità di caricare video in 3D• Possibilità di commentare video di altri utenti	E' il primo social network che ti permette di caricare video, anche in 3D, di visualizzarli ad alta qualità, di commentarli e di iscriverti a quel canale per vedere i suoi video.	Non permette la chat istantanea tra iscritti di un determinato canale.

⁹ Cfr. <http://en.wikipedia.org/wiki/Youtube>

2.6 Applicazione per i dispositivi mobili

In questo sottocapitolo andremo ad analizzare le interfacce grafiche, per i dispositivi mobili, dei tre social network più usati al momento ovvero Facebook, Google+ e Twitter. Gli altri social network, come LinkedIn e YouTube, non saranno inclusi, in quanto sono indirizzati ad un pubblico più specifico e non ad un pubblico generale come gli altri tre. Andiamo ad analizzare i problemi che un utente anziano può trovare utilizzando un'applicazione come questa sul proprio dispositivo mobile.

Sostanzialmente ci sono due problemi principali per questo tipo di applicazioni e li andremo ad analizzare per ogni interfaccia. Il primo problema che si può notare in queste interfacce è la complessità grafica con cui si presenta l'applicazione. Iniziamo da Facebook. Un utente medio-giovane sa come arrivare a visualizzare i messaggi, notifiche, inviare messaggi, richieste d'amicizia, visualizzare il proprio profilo, etc. Un utente anziano non sa fare queste cose e le uniche cose chiare per loro sono quelle evidenziate dai tre bottoni principali che hanno la funzione di:



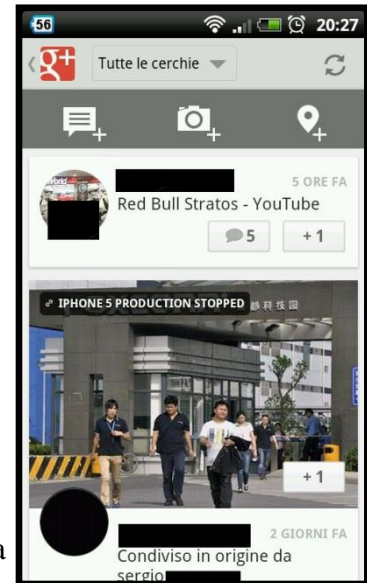
- aggiornare lo stato;
- scattare ed *uploadare* nuove foto;
- registrare la propria posizione.

Quest'ultima funzione ha il compito di far sapere, agli amici dove ci si trova al momento ma, essendo poco chiara, un utente anziano non la proverebbe nemmeno.

Da notare che, usando l'applicazione mobile di Facebook, non è possibile condividere qualcosa come foto, link, etc.

E' il turno di Google+. Vediamo che l'interfaccia è un po' più complessa rispetto a quella di Facebook ma offre molte più possibilità. I bottoni che saltano all'occhio sono sempre i soliti tre che servono alle stesse funzioni di quelli di Facebook ovvero:

- aggiornare lo stato;
- scattare ed *uploadare* nuove foto;
- registrare la propria posizione.



Ciò che Google+ inserisce rispetto a Facebook, è la possibilità di visualizzare determinati post provenienti da una cerchia che possiamo scegliere dalla *DropDownList* in alto, da tutte le nostre cerchie, oppure possiamo visualizzare semplicemente i temi caldi di cui si parla su Google+. Google+ per dispositivi mobili offre un'altra novità rispetto alla versione di Facebook: cliccando su un link o su qualcosa che si vuole condividere, lo si potrà fare senza dover accedere dal proprio portatile per condividere ciò che abbiamo notato in quel momento e che, più tardi, potrebbe essere troppo difficoltoso trovarlo per via di altri post.

L'ultima applicazione che andremo ad analizzare è Twitter. L'interfaccia grafica di Twitter è un po' più semplice anche se, il significato dei bottoni Connetti e Scopri, non è proprio chiaro. Il bottone Connetti mostra le persone che ti stanno seguendo (*followers*) e in che *tweet* sei stato citato. Il bottone Scopri ha la funzione di far vedere le tendenze di Twitter, quelle scritte con davanti l'Hashtag #, mostra i suggerimenti su chi seguire, le attività dei tuoi *followed* e le ultime notizie retwittate dai tuoi *followed*. Il pulsante Home visualizza le cose pubblicate dai tuoi *followed*, mentre il pulsante Account visualizza l'Account e le informazioni. Anche qui, come su Google+, se si vuole condividere il pensiero di un altro (*retweet*) lo si può fare in maniera istantanea.



Il secondo problema che un utente anziano incontra è un insieme di problemi derivanti dall'interfaccia e dai problemi che un anziano può avere come possibili problemi di vista, tremolio alle mani, apprendimento di cose nuove, etc. I problemi che sorgono dopo un'attenta riflessione sono i seguenti:

- grandezza dei bottoni eccessivamente piccola. Questo rende impossibile per un utente anziano mettere 'Mi piace' a dei post o link su Facebook (o mettere i '+1' se si utilizza Google+), in quanto sono piccoli e non hanno la mano ferma;
- impossibilità di commentare foto in quanto non si ha la mano ferma e si incorre nel rischio di scrivere cose sbagliate;
- difficoltà di aggiungere persone, in quanto il procedimento da fare è alquanto difficile per un utente anziano e non è immediato. Anche se un parente, o un amico gli invierà la richiesta d'amicizia, avrà difficoltà nell'accettarlo perché non saprà che fare.

A questi problemi si aggiunge anche il fatto che i caratteri, utilizzati da queste applicazioni, sono piccoli e gli anziani non riescono a leggere eventuali commenti, descrizioni di immagini, etc.

Dopo un'attenta ricerca, effettuata dal gruppo LifeParticipation, si è potuto arrivare ad una soluzione ovvero ad un'interfaccia semplice, senza bottoni, che ti permetta di visualizzare direttamente le foto dei tuoi parenti, a schermo intero, senza scervellarsi troppo. L'interfaccia ideale sarebbe quella di avere due pulsanti che occupino il lato basso dello schermo, con una faccina felice per il 'mi piace' ed una faccina triste per il 'non mi piace'. La tastiera, per un'applicazione rivolta agli anziani, è da escludere in quanto non piace agli utenti.

Capitolo 3

Progetto di ricerca Life Participation

In questo capitolo parlerò del progetto di ricerca ‘Life Participation’ illustrando quali sono le applicazioni create dal gruppo di ricerca, le applicazioni da creare e i problemi riscontrati grazie ai feedback degli utenti. Prima di tutto introduciamo il progetto Life Participation e gli obiettivi del loro.

Il gruppo di ricerca, disegna e sviluppa soluzioni che fanno vivere le persone un po’ meglio. Il punto di partenza è l’attenzione che le persone anziane necessitano in termine di benessere emotivo. Il gruppo lavora con le persone spendendo più tempo possibile sul campo piuttosto che nel laboratorio, per creare innovazioni che fanno la differenza.

Il gruppo di ricerca preferisce, invece di pubblicare i risultati che ottiene, la soddisfazione delle persone. Questo è il loro obiettivo principale.

Il gruppo è sempre più interdisciplinare e diverso, il che provvede ad un ambiente di pensiero stimolante e gratificante. Essi accolgono le cooperazioni con i gruppi da tutto il mondo come un grande valore nelle interazioni e scambio di propositi, idee e risultati.¹⁰

Gli obiettivi da raggiungere di questo gruppo e del progetto di ricerca Life Participation, sono:

- Far avvicinare le persone più anziane al nuovo mondo dei *device* mobili;
- Cercare di migliorare lo stile di vita delle persone anziane facendole divertire e rendendole anche partecipi allo sviluppo delle applicazioni;
- Sviluppare applicazioni con interfacce grafiche semplici e molto più *user-friendly* di quelle esistenti che sono pensate più per un pubblico giovane.

¹⁰ Cfr. <http://www.lifeparticipation.org/aboutus.html>

Per raggiungere questi obiettivi il gruppo di ricerca, prima di modificare qualcosa, chiede il parere agli anziani rendendoli così partecipi in modo da farli avvicinare un po' di più al mondo tecnologico, che ormai sta prendendo sempre più piede.

Le persone anziane si sentono più invogliate ad imparare entrando a far parte di un progetto ed aumenta anche la loro voglia di affrontare queste nuove sfide (il mondo tecnologico) se sono affiancati da altre persone che, con calma, gli spiegano i passi da fare per effettuare un'azione.

3.1 Applicazioni sviluppate

In questo sottocapitolo, andrò ad analizzare le applicazioni sviluppate sia per *device* mobili e per computer, dal gruppo di ricerca del progetto Life Participation.

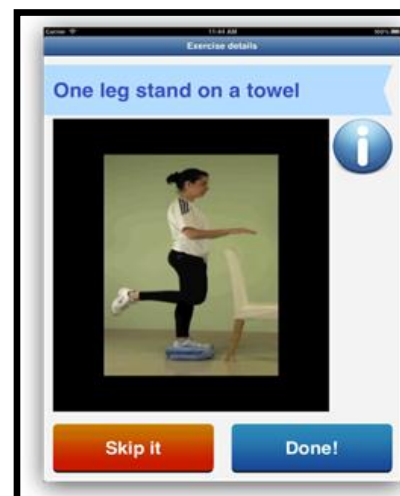
Nonostante gli obiettivi del gruppo possano far pensare che le applicazioni sono rivolte solo ed esclusivamente ad un pubblico di anziani, non è così. Difatti certe applicazioni hanno come *target* principale di utenza le persone anziane, altre applicazioni hanno il *target* rivolto alle persone giovani, mentre altre applicazioni sono rivolte sia agli anziani che alle persone giovani.

Andiamo ad elencare e ad analizzare nello specifico, le applicazioni da loro sviluppate:

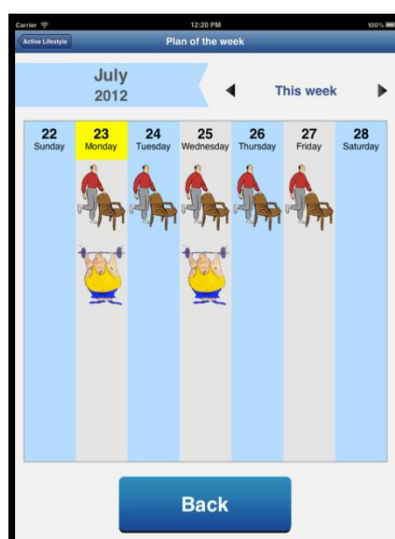
- *Active Lifestyle;*
- *Active Brain;*
- *Come Along;*
- *Liquid Galleries;*
- *Whats Up?.*

3.1.1 Active Lifestyle

Quest'applicazione è nata in quanto, l'*US Census Bureau* ha riferito che in 10 anni le persone anziane saranno più numerose rispetto ai bambini al di sotto di 5 anni e l'Europa sarà il continente più 'grigio'. Il gruppo di ricerca del progetto Life Participation ha sviluppato l'applicazione per iPad *Active Lifestyle* per supportare al meglio il processo di invecchiamento. L'applicazione *Active Lifestyle* offre un piano di esercizi di allenamento dedicati sulla forza e sull'equilibrio, per gli anziani. Questo piano



Dimostrazione di un esercizio



Un esempio di piano settimanale

personalizzato permette agli anziani di diventare più autonomi, migliorare la loro salute e anche la loro qualità della vita. Usando un'interfaccia amichevole e facile da usare, gli anziani possono ricevere un suono di allerta quando un nuovo esercizio deve essere svolto. In aggiunta, loro controllano i dettagli di ogni esercizio pianificato, salvano la loro *performance* e controllano i progressi. Gli anziani possono anche condividere i propri risultati con operatori sanitari (ad esempio medici, fisioterapisti, etc..), amici (ad esempio gli anziani che usano la stessa applicazione) e parenti al fine di motivarli a migliorare ed a raggiungere risultati sempre migliori.¹¹

¹¹ Cfr. <http://www.lifeparticipation.org/active.html>

3.1.1.1 Riassumendo:

Target principale di utenti	Sviluppato per...	Servizi offerti	Aspetti positivi
Persone anziane	<ul style="list-style-type: none">• iOS.(iPad)	<ul style="list-style-type: none">• Piano di esercizi di allenamento dedicati sulla forza e sull'equilibrio (personalizzato)• Possibilità di condividere i propri risultati con conoscenti, fisioterapisti, amici e parenti	<ul style="list-style-type: none">• Interfaccia semplice ed amichevole• Miglioramento del proprio equilibrio, autonomia e del proprio stile di vita

3.1.2 Active Brain

Quest'applicazione ha come *target* principale di utenti gli anziani, in quanto mira a migliorare i propri riflessi. *Active Brain* ha preso spunto dal classico gioco arcade *Dance Dance Revolution* modificandolo in modo da renderlo più facile per gli anziani.

Lo scopo del gioco è quello di riuscire a prendere tutti i simboli che cadono dall'alto dello schermo fino a raggiungere il basso dove ci sono i bottoni da premere. I bottoni hanno le immagini dei semi delle carte da ramino, ovvero picche(♠), fiori(♣), cuori(♥) e quadri(♦). Quando l'oggetto è finito sotto al seme da premere, bisogna premere su quel bottone e si guadagneranno punti.

Le canzoni usate sono quelle da sala da ballo in modo da essere godibili all'orecchio delle persone anziane. Alla fine della canzone verrà mostrato il punteggio in percentuale.¹²



Interfaccia del gioco durante l'esecuzione

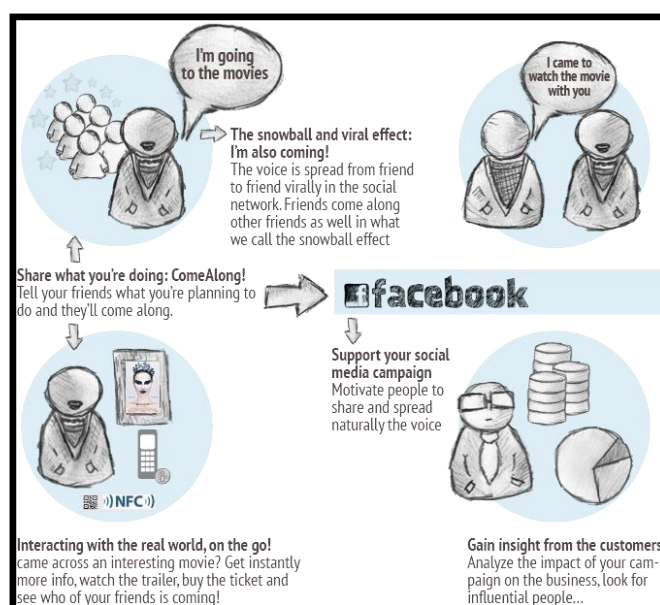
¹² Cfr. <http://www.lifeparticipation.org/activeBrain.html>

3.1.2.1 Riassumendo:

Target principale di utenti	Sviluppato per...	Servizi offerti	Aspetti positivi
Persone anziane	<ul style="list-style-type: none"> iOS.(iPad) Web 		<ul style="list-style-type: none"> Interfaccia semplice ed amichevole Miglioramento dei propri riflessi Belle musiche

3.1.3 Come Along

Al giorno d'oggi i *social networks* permettono alle persone di stare in contatto sia con quelle che vivono nel vicinato e sia con le persone che sono lontano. In questa maniera le relazioni sociali sono mediate dai computer, mentre nella vita reale le esperienze sociali hanno molto più valore nella nostra vita.



Funzionamento concettuale di ComeAlong



Visualizzazione delle proiezioni

In questo contesto, *ComeAlong*, mira a motivare le persone a partecipare assieme ad eventi sociali come andare a vedere un film al cinema, andare ad una lezione di ginnastica, bere una birra in un pub, andare ad un concerto. Condividere le proprie intenzioni attraverso un social network, permette agli amici dell'utente di essere a conoscenza degli eventi e della partecipazione di qualcuno che loro conoscono. In questa maniera le persone possono venire ed avere una migliore esperienza grazie alla partecipazione dei propri amici.¹³

¹³ Cfr. <http://www.lifeparticipation.org/comealong.html>

ComeAlong ti permette di visualizzare a quali film i tuoi amici vanno, cliccando sul *tab* ‘Amici’, in modo da poterti prenotare ed andare assieme a loro. Hai la possibilità di comperare il biglietto del cinema per il film da te scelto. Tutto questo avviene tramite una procedura sicura e tramite l’utilizzo del codice QR il quale indica quale posto hai comperato, per che film e di che orario. Il codice QR può essere letto con una delle tante applicazioni disponibili sul *market* oppure attraverso la tecnologia NFC.

3.1.3.1 Riassumendo:

Target principale di utenti	Sviluppato per...	Servizi offerti	Aspetti positivi
Persone giovani	<ul style="list-style-type: none"> • iOS • Android • Web (FB App) 	<ul style="list-style-type: none"> • Possibilità di prenotare e comperare il posto alla proiezione scelta • Possibilità di condividere su FaceBook che film si va a vedere e a che cinema • Possibilità di vedere che film vanno a vedere i tuoi amici ed, eventualmente, raggiungerli • Possibilità di commentare un film o di mettere un mi piace • Possibilità di leggere i commenti altrui sul film 	<ul style="list-style-type: none"> • Interfaccia semplice ed intuitiva • Prenotazione/comperare del posto al film scelto • Non necessita di una registrazione (login FaceBook)

3.1.4 Liquid Galleries

Il gruppo di ricerca del progetto *Life Participation* crede che ogni museo dovrebbe provvedere ad un'unica esperienza, che deve essere personalizzata e condivisa attraverso il pubblico. Il visitatore dovrebbe avere il pieno controllo della sua esperienze, dovrebbe avere accesso a tutte le conoscenze relative ad una cosa specifica. L'applicazione *Liquid Galleries* permette questa cosa. Usando la tecnologia mobile e le conoscenze di internet, il visitatore è supportato durante la sua permanenza grazie a quest'applicazione.

Un oggetto in una mostra non è godibile solo attraverso la sua rappresentazione fisica o da una breve descrizione su un'etichetta, ma anche attraverso i suoi eventi storici, attraverso la ricerca fatta su di esso durante questi anni, attraverso i commenti e i pensieri di scienziati e studenti, e attraverso la comunità di persone interessate ad esso.



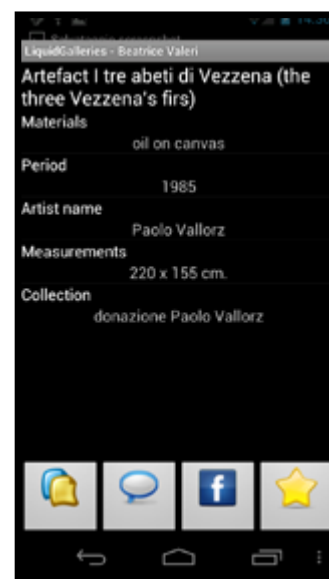
Interfaccia principale

Liquid Galleries da un vasto supporto e una linea guida ai visitatori, dando loro informazioni sugli oggetti esposti e sulle questioni connesse del museo. In aggiunta ai commenti della comunità e ai contributi all'artefatto, *Liquid Galleries*, permette anche agli scienziati di condividere la loro ricerca basata sulla conoscenza associata ai musei e i musei, a loro volta, di condividere le informazioni dei loro cataloghi. Finché la visita al museo è dedicata all'esperienza di oggetti reali, le informazioni sono provvedute anche una volta tornati a casa, quando si ha più tempo per entrare nello specifico.

La conoscenza è disponibile per i visitatori ad ogni tempo così loro possono catturare e condividere con gli amici e colleghi, le informazioni in cui loro sono interessati. In questa maniera, durante la visita, le persone possono concentrarsi sull'esperienza di oggetti reali senza perdere la possibilità di entrare nello specifico.¹⁴

¹⁴ Cfr. <http://www.lifeparticipation.org/liquid.html>

L'interfaccia di *Liquid Galleries* si presenta con 4 *tab*. Il primo *tab* 'Welcome' ti permette di fare il logout da FaceBook, di ricevere informazioni sull'opera leggendo il code QR vicino a quest'ultima oppure usando la tecnologia NFC. Nel secondo *tab* 'Top Artifact' ti permette di leggere i titoli delle opere che sono state analizzate attraverso il QR o NFC. Cliccando su una di queste opere è possibile vedere il quadro, leggere tutte le sue informazioni e condividere su FaceBook, aggiungere ai preferiti, aggiungere un commento e vedere i commenti effettuati dagli altri utenti. Il terzo *tab* 'Fast Bookmark' permette di attivare una funzione che ti aggiunge, facilmente e velocemente, interessanti artefatti ai propri favoriti. Una volta cliccato sul bottone 'Start fast bookmark', basterà leggere il codice QR o il *tag* NFC relativo ad uno specifico artefatto, per aggiungerlo automaticamente alla lista dei propri favoriti. L'ultimo *tab* è 'Personal Page' il quale mostrerà le vostre informazioni personali.



Informazioni di un'opera

3.1.4.1 Riassumendo:

Target principale di utenti	Sviluppato per...	Servizi offerti	Aspetti positivi
Persone giovani ed anziani	<ul style="list-style-type: none"> Android Web 	<ul style="list-style-type: none"> Possibilità di salvare l'immagine dell'opera Possibilità di visualizzare e commentare l'opera Possibilità di aggiungere l'opera ai preferiti Possibilità di condividere l'opera su FaceBook 	<ul style="list-style-type: none"> Interfaccia semplice ed intuitiva Si ricevono le informazioni relative alle opere e si possono visualizzare offline Si può visualizzare le opere offline

3.1.5 What's Up?

What's Up? è una piattaforma di comunicazione mobile che promuove l'interazione intergenerazionale tra anziani e i loro parenti più giovani e amici. Da una parte l'utente giovane usa *What's Up?*, che è un'applicazione con più funzionalità sviluppate per il dispositivo di comunicazione più utilizzato dai giovani, ovvero lo *smartphone*. Dall'altra parte gli utenti più anziani, usano *What's Up? Display*, che è un'applicazione sviluppata per i *tablet* (iPad) che lavora inizialmente, come visualizzatore dei messaggi e delle immagini ricevute dall'applicazione *What's Up?*, adattando la sua interfaccia per poter supportare anche la condivisione proveniente dall'anziano verso il più giovane.

E' un fatto della vita che, crescendo, le nostre capacità sono ridotte sotto diversi aspetti come, la diminuzione della vista, dell'udito e delle abilità psicomotorie.¹⁵ A causa di questi cambiamenti, l'interazione degli anziani con i parenti e amici, tende a ridursi, spesso ridotta a delle sporadiche chiamate e gli incontri con i parenti di solito sono sempre proposti dagli anziani. Molti nonni parlano ai loro nipoti meno di una volta ogni due settimane, le chiamate raramente durano più di cinque minuti e i nonni (o i genitori) iniziano la maggior parte delle chiamate.¹⁶



WhatsUp? interfaccia invio messaggio

Dall'altra parte, le relazioni sociali sono indubbiamente una dei fattori più importanti che influenzano il benessere ed i problemi che ho menzionato precedentemente, possono solamente ostacolare l'abilità nel mantenere e a nutrire dei rapporti con amici e parenti. Le tecnologie dell'informazione e della comunicazione possono contribuire ad affrontare questo problema considerando meglio sia le capacità degli anziani e i modi di comunicazione dei giovani. Seguendo questi requisiti generali, *What's Up?* è una piattaforma di comunicazione mobile, progettata specificamente per favorire l'interazione sociale tra gli utenti anziani e i loro amici e la famiglia, rompendo l'isolamento degli anziani.¹⁷

¹⁵ Cit. Drolet et. al. 2010, Fozard & Salant 2001

¹⁶ Cit. Evjemo 2004

¹⁷ Cfr. <http://www.lifeparticipation.org/whatsup.html>

What's Up? Display è stato sviluppato con tre possibili interfacce per gli anziani:

- la prima interfaccia permette solamente di visualizzare le novità proveniente dai contatti;
- la seconda interfaccia ci dà la possibilità di mettere mi piace o non mi piace ai messaggi/immagini che visualizziamo;
- la terza interfaccia è quella più completa. Essa ci dà la possibilità di mettere mi piace o non mi piace ai messaggi/immagini che si visualizzano e di commentare.



Interfacce WhatsUp? Display

Il gruppo di ricerca del progetto Life Participation sta sviluppando ancora quest'applicazione, aggiungendo nuove cose come la sincronizzazione tra i vari dispositivi mobili ma, di questo, ne parleremo nel prossimo capitolo.

3.1.5.1 Riassumendo:

Target principale di utenti	Sviluppato per...	Servizi offerti	Aspetti positivi
Persone giovani ed anziani	<ul style="list-style-type: none"> Android (<i>What's Up?</i>) iOS (<i>What's Up? Display</i> & <i>What's Up?</i>) 	<ul style="list-style-type: none"> Possibilità di cambiare interfaccia (<i>What's Up? Display</i>) Possibilità di commentare, mettere mi piace o non mi piace su immagini/messaggi ricevuti (<i>What's Up? Display</i>) Possibilità di inviare immagini e messaggi ai propri contatti (<i>What's Up?</i>) Possibilità di invitare amici ad usare l'applicazione (<i>What's Up?</i>) Possibilità di sincronizzare immagini e messaggi tra vari <i>device</i>. 	<ul style="list-style-type: none"> Interfaccia semplice ed intuitiva Relazioni migliorate tra gli anziani ed i loro parenti, amici Tre interfacce differenti che possono essere scelte dal l'utente a seconda delle sue capacità (<i>What's Up? Display</i>)

Capitolo 4

Sincronizzazione di risorse tra dispositivi Android

In questo capitolo parlerò della sincronizzazione di risorse tra dispositivi Android andando ad illustrare i metodi da me utilizzati per implementarlo, i vari *Use Case Diagram*, i vari *Activity Diagram* utilizzati come protocollo per scambiare messaggi tra i dispositivi coinvolti e delle immagini relative all'applicazione sviluppata.

La sincronizzazione di risorse tra dispositivi Android, che d'ora in poi chiamerò solamente sincronizzazione, è stata sviluppata per l'applicazione *What's Up?* cosicché potesse colpire un *range* di utenti più ampio dato che nessun social network sui *device* mobili, permette di avere una visualizzazione in contemporanea tra due utenti delle immagini condivise tra di loro.

Prima di parlare delle tecnologie utilizzate per lo sviluppo di questa *feature*, parliamo in linee generali del sistema operativo Android.

4.1 Introduzione ad Android

Android è un sistema operativo basato su *Linux* progettato principalmente per dispositivi mobili *touchscreen* come *smartphones* e *tablet*. Inizialmente sviluppato da Android Inc. finanziata da Google e poi comperata nel 2005, Android è stato inaugurato nel 2007 con la fondazione del *Open Handset Alliance*: ovvero di un consorzio di aziende *hardware*, *software* e di telecomunicazione devote a promuovere ed a sviluppare degli standard per dispositivi mobili. Il primo telefono con sistema operativo Android, è stato venduto nell'ottobre del 2008.

Android è *open source* e Google ha rilasciato il codice sotto licenza *Apache*. Questo codice *open source* e le licenze permettono al software di essere modificato liberamente e distribuito dai produttori di dispositivi, operatori wireless e degli sviluppatori. Inoltre, Android ha una grande comunità di sviluppo che scrive applicazioni (“*apps*”) che estendono la funzionalità dei dispositivi, scritto principalmente in una versione personalizzata di *Java*. Nell'ottobre 2012 erano

disponibili all'incirca 700.000 *apps* per Android, e il numero di applicazioni scaricate da *Google Play*, che è l'*app store* principale di Android, era di 25 bilioni.

Questi fattori hanno permesso ad Android di diventare la piattaforma più utilizzata del mondo per quanto riguarda gli *smartphones* ed anche il software che viene scelto dalle aziende che richiedono un costo basso, personalizzazione, leggerezza del sistema operativo per i dispositivi *high tech* senza perdere tempo a svilupparne uno da zero. Come conseguenza, pur essendo progettato principalmente per telefoni e *tablet*, si è potuto notare l'aggiunta di applicazioni su televisioni, console di gioco ed altri oggetti elettronici. La natura libera di Android ha incoraggiato una grande comunità di sviluppatori ed appassionati, ad utilizzare il codice *open source* come base per i loro progetti, che aggiunge nuove *features* per gli utenti avanzati o portare Android sui dispositivi nei quali sono presenti altri sistemi operativi.¹⁸

4.2 Tecnologie utilizzate

In questo sottocapitolo, andrò ad analizzare le tecnologie utilizzate per sviluppare l'applicazione adatta alla sincronizzazione di risorse tra dispositivi android.

Le tecnologie utilizzate per sviluppare l'applicazione sono:

- MQTT (*MQ Telemetry Transport*);
- C2DM (*Cloud to Device Messaging*);
- GCM (*Google Cloud Messaging*);
- Uniqush.

Nei seguenti sottocapitoli, parlerò di queste tecnologie anche se sono state scartate per motivi tecnici oppure sostituite per via di aggiornamenti introducendole e dando uno sguardo ai loro vantaggi e svantaggi.

¹⁸ Cfr. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

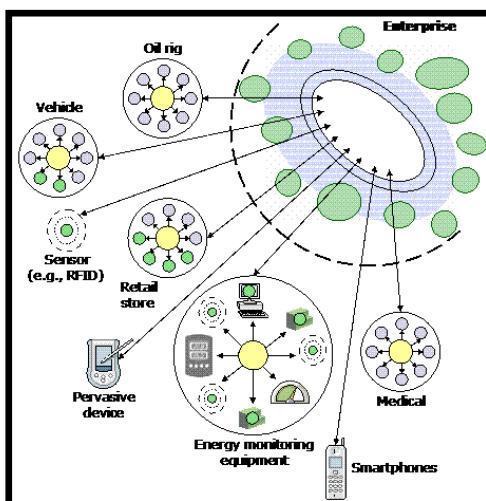
4.2.1 MQTT (*MQ Telemetry Transport*)

MQTT è un protocollo estremamente semplice e leggero, disegnato per dispositivi con banda bassa, ad alta latenza o reti inaffidabili. Gli scopi principali sono quello di minimizzare la latenza della rete, tentare di garantire affidabilità ed un certo grado di sicurezza nella consegna. Questi principi rendono il protocollo un ideale per l'emergente mondo del “*machine-to-machine*” (M2M), e per le applicazioni mobili dove la banda e la durata della batteria sono importanti.

MQTT non ha uno standard formalizzato, ma il protocollo è reso pubblico e delle compagnie come Eurotech (conosciuta anche come Arcom) ha implementato il protocollo nei loro prodotti.

Nel novembre 2011 IBM ed Eurotech hanno annunciato la loro partecipazione nel *Eclipse M2M Industry Working Group* e la donazione del codice MQTT al progetto proposto *Eclipse Paho*.

La gestione con gli altri dispositivi viene gestita tramite l'utilizzo del protocollo MQTT usando due porte TCP/IP. La prima porta è la 1883 ed è riservata con IANA per l'uso del protocollo MQTT mentre la porta 8883 è riservata all'uso del protocollo MQTT tramite SSL.



Esempio di MQTT M2M

MQTT supporta la sicurezza in quanto si può inviare username e password tramite il protocollo che gestisce l'encrittazione nella rete che può essere gestita tramite SSL, indipendentemente dal protocollo MQTT. Altra sicurezza può essere aggiunta da un'applicazione che encrypta i dati che invia e che riceve, ma questo non è incluso nel protocollo MQTT, in quanto non lo manterrebbe semplice e leggero.¹⁹

¹⁹ Cfr. <http://mqtt.org/faq>

4.2.1.1 Riassumendo:

Vantaggi	Svantaggi
<ul style="list-style-type: none">• Connessioni stabilite tramite porta TCP/IP• L'invio e la ricezione dei messaggi è assicurata in quanto si è creato un canale di comunicazione tra i due dispositivi• Affidabile• Facile da implementare• Molta sicurezza già impostata di default• Leggero e semplice	<ul style="list-style-type: none">• Non si può usare questo protocollo se la rete a cui si è connessi, ha un <i>proxy</i>.

A causa del fatto che MQTT non può essere utilizzato dietro *proxy*, questa soluzione è stata scartata in quanto l'applicazione deve girare senza problemi anche dietro *proxy*. E' per questo che la prima opzione di sviluppo è stato C2DM, seguito poi da GCM suo successore, in quanto riesce ad inviare messaggi utilizzando la porta 80.

4.2.2 C2DM (*Cloud To Device Messaging*)

In questo sottocapitolo elencherò le caratteristiche principali di C2DM ed effettuerò una panoramica sull'architettura.

Le caratteristiche principali di C2DM e le modalità di utilizzo sono:

- Può essere utilizzato per inviare messaggi da parte di server alle loro applicazioni Android. Il servizio di messaggistica non è disegnato per inviare tanti contenuti agli utenti via messaggio. Preferibilmente, dovrebbe essere usato per dire alle applicazioni che ci sono nuovi dati sul server, così le applicazioni possono prelevarle;
- C2DM non garantisce la consegna o l'ordine dei messaggi;
- Un'applicazione su un dispositivo Android non deve andare per ricevere i messaggi. Il sistema sveglierà l'applicazione tramite un *broadcast Intent* quando il messaggio arriva;
- Non prevede nessuna interfaccia precostruita al suo interno oppure la possibilità di manipolare i messaggi. C2DM passa semplicemente il messaggio *raw* ricevuto direttamente all'applicazione, che ha il pieno controllo su come maneggiarla;

- Richiede Android 2.2 o maggiore per poter funzionare;
- Usa già una connessione esistente con i servizi Google. Questo richiede agli utenti di impostare il loro Google account sui loro dispositivi mobili.

Dopo aver dato uno sguardo alle caratteristiche principali di C2DM, passiamo a dare una panoramica sull'architettura. Possiamo distinguere due categorie principali per l'architettura che sono:

- Componenti: sono le entità fisiche che hanno un ruolo in C2DM;
- Credenziali: sono gli IDs e i *tokens* che sono usati nelle differenti parti di C2DM per assicurare che tutte le parti sono state autenticate e che il messaggio sta andando nel posto corretto.

La categoria dei componenti si divide in tre punti principali e sono:

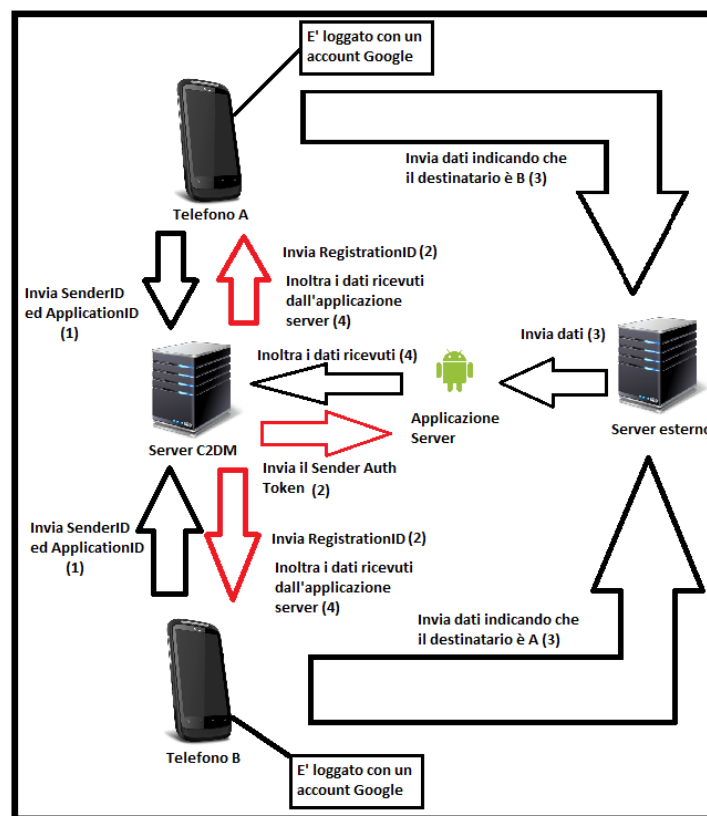
- Dispositivo mobile: il dispositivo sul quale sta andando l'applicazione Android che usa C2DM. Esso deve essere un dispositivo che supporta Android 2.2 e che abbia il *Market* installato, e deve avere almeno un account Google loggato;
- Server esterno: un server che gli sviluppatori hanno impostato come parte dell'implementazione di C2DM nelle loro applicazioni (a questo scopo ho usato Uniquish i cui dettagli verranno spiegati successivamente). Questo server invia i dati da un'applicazione Android al dispositivo tramite il server C2DM;
- Server C2DM: i server di Google sono coinvolti nel ricevere i messaggi dai server esterni ed inviarli ai dispositivi mobili.

La categoria delle credenziali si divide in 5 punti principali e sono:

- *Sender ID*: ovvero un'email account associata all'applicazione dello sviluppatore. Il *sender ID* è usato nel processo di registrazione per identificare l'applicazione Android alla quale è permesso inviare messaggi al dispositivo;
- *Application ID*: ovvero l'applicazione che viene registrata per ricevere i messaggi. L'applicazione è identificata dal nome del *package* dal *manifest*. Questo assicura che i messaggi sono indirizzati all'applicazione corretta;

- *Registration ID*: un Id rilasciato dai server C2DM all'applicazione Android il quale permette di ricevere messaggi. Una volta che l'applicazione ha il *registration ID*, lo invia all'applicazione server che lo usa per identificare ogni dispositivo che è registrato a ricevere i messaggi per una determinata applicazione;
- *Google User Account*: per permettere a C2DM di funzionare il dispositivo mobile deve includere almeno un account Google loggato;
- *Sender Auth Token*: esso è un *ClientLogin Auth token* salvato sull'applicazione server che gli dà l'autorizzazione ad accedere ai servizi di Google. Il *token* è incluso nell'*header* della richiesta POST che invia il messaggio.²⁰

Ecco un'immagine che spiega il funzionamento di C2DM:



Prima di tutto i telefoni A e B, loggati con un account Google, inviano il *SenderId* e l'*ApplicationId* al Server C2DM il quale invierà a sua volta il *RegistrationID* all'applicazione presente nel telefono ed allo stesso tempo, invierà il *Sender Auth Token* all'applicazione server. Se il telefono A vuole comunicare con B, esso invia i dati al server esterno indicando che il destinatario è B. Il server invia i dati ricevuti

²⁰ Cfr. <https://developers.google.com/android/c2dm/>

all'applicazione server la quale contatta il server C2DM che proverà a consegnarli al destinatario esatto, riconosciuto dal *RegistrationID*. Non è assicurata la consegna in quanto C2DM non si preoccupa se un messaggio viene consegnato o meno in quanto non è adatto ad un continuo invio di messaggi ma, piuttosto, ad un invio di messaggi moderato (es: una nuova versione disponibile).

4.2.2.1 Riassumendo:

Vantaggi	Svantaggi
<ul style="list-style-type: none">• <i>RegistrationID</i> univoci per ogni dispositivo che utilizza quell'applicazione• API ben commentate e disponibili• Facile da implementare• Leggero e semplice• Si può usare dietro <i>proxy</i> in quanto inviano i messaggi in POST	<ul style="list-style-type: none">• Non è assicurato l'arrivo dei messaggi se continui• Lunghezza massima dei messaggi di 140 caratteri

Nonostante non sia assicurato l'arrivo dei messaggi se essi sono continui, ho optato per implementare questa scelta. Essendo in beta il 26 giugno 2012, è stato sostituito da GCM (*Google Cloud Messaging*) che vado ad illustrare subito.

4.2.3 GCM (*Google Cloud Messaging*)

In questo sottocapitolo elencherò solo le caratteristiche principali di GCM in quanto l'architettura è pressoché uguale al suo predecessore C2DM.

Le caratteristiche principali di GCM e le modalità di utilizzo sono:

- Può essere utilizzato per inviare messaggi da parte di server alle loro applicazioni Android. Con GCM non si ha il limite dei 1024 caratteri come lo aveva C2DM. I messaggi inviati vengono consegnati direttamente al telefono destinatario:
- Un'applicazione su un dispositivo Android non deve andare per ricevere i messaggi. Il sistema sveglierà l'applicazione tramite un *broadcast Intent* quando il messaggio arriva;

- Non prevede nessuna interfaccia precostruita al suo interno oppure la possibilità di manipolare i messaggi. GCM passa semplicemente il messaggio *raw* ricevuto direttamente all'applicazione, che ha il pieno controllo su come maneggiarla;
- Richiede Android 2.2 o maggiore che ha l'applicazione *Google Play Store* installata, od un emulatore su cui gira Android 2.2 con le API di Google;
- Usa già una connessione esistente con i servizi Google. Per i dispositivi con la versione di Android inferiore alle 3.0, richiede agli utenti di impostare il loro Google account sui loro dispositivi mobili. Sui dispositivi mobili sul quale gira Android 4.0.4 o maggiore, non è richiesto il login del loro Google account.²¹

La panoramica architetturale è pressoché identica a quella del suo predecessore C2DM quindi non la andrò a spiegare. Basti sapere che al posto dei C2DM Servers, ci sono i GCM Servers per il resto è tutto uguale. Anche il funzionamento è lo stesso (v.di immagine precedente).

Andiamo ad illustrare le differenze tra C2DM e GCM:

- Facile da usare. Per far funzionare C2DM su un dispositivo bisognava registrarsi ora non più;
- Aumentata l'efficienza della batteria;
- GCM provvede a delle librerie d'aiuto sia per client che per server;
- Nuove API;
- *Simple API Key*: per usare il servizio GCM, bisogna ottenere una *Simple API Key* dalle API di Google. GCM accetta solo le *Simple API Key* quindi usando il *ClientLogin* o *oAuth2 tokens* non funzionerà;
- *Sender ID*: in C2DM il *Sender ID* è un indirizzo e-mail. In GCM, il *Sender ID* è un numero del progetto che può essere acquisito dalle API console;
- Formato JSON: le richieste HTTP di GCM supportano il formato JSON in aggiunta al testo normale (*plain text*);

²¹ Cfr. <http://developer.android.com/google/gcm/gcm.html>

- Messaggi multicast: in GCM puoi inviare lo stesso messaggio a più dispositivi in contemporanea. Per esempio, un'applicazione di sport vuole aggiornare il punteggio di una partita. Prima con C2DM non poteva ora con GCM, può inviare lo stesso messaggio a più di 1000 *Registration ID* nella stessa richiesta (questo richiede JSON);
- Tempo di vita dei messaggi (*time-to-live*): applicazioni come video chat e calendario, possono inviare degli inviti a degli eventi con un valore di *time-to-live* tra 0 e 4 settimane. GCM immagazzinerà i messaggi finché non scade il *time-to-live*. Un messaggio con un valore del *time-to-live* pari a 0 non verrà immagazzinato nel server GCM;
- Messaggi con del “carico”: le applicazioni possono usare messaggi con del “carico” per consegnare messaggi della dimensione fino a 4 Kb. Questo può essere utile nelle applicazioni chat;
- Canonici *Registration ID*: ci potrebbero essere delle situazioni dove il server salva 2 *Registration ID* per lo stesso dispositivo. Se la risposta GCM contiene un *Registration ID*, semplicemente rimpiazza il *Registration ID* vecchio con quello che si è provveduto. Con questa *feature* l'applicazione non deve inviare il *Device ID* al server.

C2DM e GCM non sono interoperabili. Per esempio non puoi postare le notifiche da GCM al *Registration ID* di C2DM, né usare il *Registration ID* di C2DM come se fosse un *Registration ID* di GCM. L'applicazione server deve tenere traccia se il *Registration ID* viene da C2DM o GCM.²²

²² Cfr. <http://developer.android.com/google/gcm/c2dm.html>

4.2.3.1 Riassumendo:

Vantaggi
<ul style="list-style-type: none">• <i>RegistrationID</i> univoci per ogni dispositivo che utilizza quell'applicazione• API ben commentate e disponibili• Facile da implementare• Leggero e semplice• Si può usare dietro <i>proxy</i> in quanto inviano i messaggi in POST• Si può inviare lo stesso messaggio a più persone in contemporanea usando JSON• Lunghezza dei messaggi illimitata• Possibilità di installare un <i>time-to-live</i> al messaggio• Possibilità di inviare messaggi con del carico massimo di 4Kb• I messaggi vengono sempre consegnati

4.2.4 Uniqush

Uniqush è un programma *stand alone*. Esso ascolta su una porta definita all'interno del file di configurazione una volta avviato. Per inviare una richiesta ad Uniqush, basta connettersi semplicemente alla porta ed inviare una richiesta HTTP Post con i parametri all'interno del *body*.

Per poter far comunicare due dispositivi tra di loro bisogna, prima di tutto, creare un collegamento tra di loro. Questo è dato dal *Push Service Provider*. Per creare questo servizio, bisognerà definire vari parametri che sono:

- *service*: il nome che questo servizio avrà;
- *pushservicetype*: esso può essere GCM (per i servizi utilizzati da Android) oppure APNS (per i servizi utilizzati da iOS);
- *projectid*: è l'ID del progetto che usa GCM dato da Google al momento della creazione (esclusiva GCM);
- *apikey*: è la API key dell'applicazione che userà GCM al momento della registrazione di essa su Google (esclusiva GCM);
- *cert*: è il percorso assoluto per arrivare al certificato, in formato .pem, che deve essere attribuito all'applicazione (esclusiva iOS);
- *key*: è il percorso assoluto per arrivare al file delle chiavi private, in formato .pem, che deve essere attribuito all'applicazione (esclusiva iOS);

- *sandbox*: è un campo opzionale. Se è true verrà utilizzato *sandbox* altrimenti no.

Da notare il fatto che un service può essere solo GCM o solo APNS però possono esistere due *service* con lo stesso nome ma con *pushservicetype* differenti. Uniqush sa se il mittente ha il sistema operativo iOS oppure Android e quindi riesce a reindirizzare la richiesta al *service* corretto.

Dopo aver creato il *service* bisogna che il dispositivo si sottoscriva ad un servizio effettuando una *subscribe*. I parametri per effettuare una *subscribe* sono:

- *service*: il nome del servizio che verrà usato;
- *subscriber*: ovvero il *topic* al quale colui che effettua la sottoscrizione si collega. Se io voglio contattare A che è iscritto al *topic* AChannel devo scrivere indicando che il destinatario è AChannel;
- *pushservicetype*: esso può essere GCM (per i servizi utilizzati da Android) oppure APNS (per i servizi utilizzati da iOS);
- *account*: l'account Google di colui che effettua la sottoscrizione che non deve essere uguale al *sender ID* (esclusiva GCM);
- *regid*: questo è il *Registration ID* dell'applicazione (esclusiva GCM);
- *devtoken*: *token* del dispositivo (esclusiva iOS).

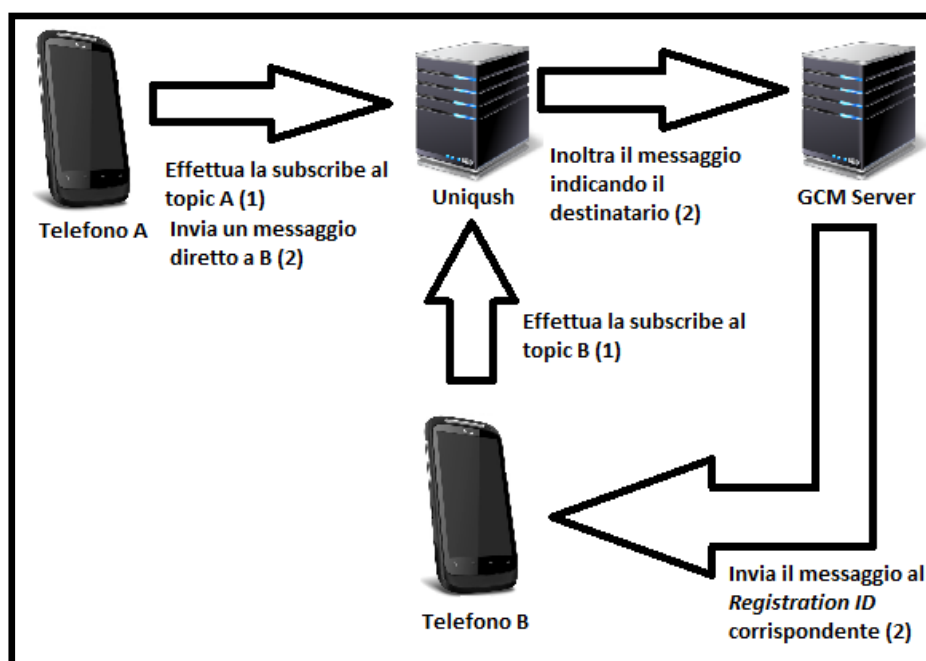
Per poter contattare un altro dispositivo, bisogna effettuare una connessione tra l'applicazione ed Uniqush attraverso HTTP. Una volta stabilita questa connessione basterà pushare il messaggio ed indicare a che dispositivo vogliamo inviarla. I parametri per il *push* sono:

- *service*: il nome del servizio che viene usato dal mittente;
- *subscriber*: il nome del destinatario. Si può inviare lo stesso messaggio a più destinatari, separandoli da delle virgole. Se si usa il carattere *, il messaggio verrà inviato a tutti i dispositivi sottoscritti;
- *msg*: al suo interno va inserito il messaggio che si vuole inviare.

Per evitare continue iscrizioni da parte dello stesso dispositivo allo stesso *topic* una volta chiusa l'applicazione, conviene effettuare un *unsubscribe*. Fare un *unsubscribe* significa non ricevere i messaggi mentre è offline e quindi di non attivare un *broadcast Intent* una volta attiva la connessione ad internet. I parametri per effettuare un *unsubscribe* sono:

- *service*: il nome del servizio usato;
- *subscriber*: ovvero il *topic* dal quale verrà effettuata la disiscrizione di quel dispositivo;
- *pushservicetype*: esso può essere GCM (per i servizi utilizzati da Android) oppure APNS (per i servizi utilizzati da iOS);
- *account*: l'account Google di colui che effettua la disiscrizione che non deve essere uguale al *sender ID* (esclusiva GCM);
- *regid*: questo è il *Registration ID* dell'applicazione (esclusiva GCM).
- *devtoken*: *token* del dispositivo (esclusiva iOS).²³

Di seguito riporto un'immagine che ci aiuterà a capire meglio il funzionamento di Uniqush:



²³ Cfr. <http://uniqush.org/documentation/usage.html>

Nell'immagine non ho messo la creazione del *service* in quanto non è di competenza dell'applicazione ma va creato a monte tramite linea di comando. Effettuata la creazione del *service*, i due telefoni effettuano una *subscribe* al *topic* a loro designato. Se A vuole contattare B, esso effettua una *push* e passerà ad Uniqush, tutti i dati necessari il quale preleverà il *RegistrationID* del destinatario fornitogli da quest'ultimo al momento della *subscribe*, contatterà il server GCM al quale inoltrerà il messaggio ed il relativo *RegistrationID* e, in seguito, esso invierà il messaggio al dispositivo con il relativo *RegistrationID*.

4.2.4.1 Riassumendo:

Vantaggi
<ul style="list-style-type: none"> • I messaggi vengono sempre consegnati al dispositivo corretto • Accoppia al topic il relativo o i relativi, <i>Registration ID</i> dei dispositivi • Può inoltrare lo stesso messaggio a più dispositivi in contemporanea • Non serve ricreare il service ogni volta • Viene utilizzato in automatico il server corretto (GCM per dispositivi Android, APNS per dispositivi Apple) • Costantemente aggiornato • Supporto immediato per eventuali problemi • Si possono creare più service con lo stesso nome che utilizzano servizi diversi (APNS e GCM) • Si può effettuare il <i>subscribe</i> a più <i>topic</i> • Una volta effettuato l'<i>unsubscribe</i> il <i>RegistrationID</i> di quel dispositivo verrà tolto e non riceverà i messaggi che vengono inviati quando non è <i>online</i> • Si può effettuare la <i>subscribe</i> ad un <i>topic</i> a cui precedentemente, si era effettuato l'<i>unsubscribe</i> • L'amministratore può mandare messaggi a coloro che utilizzano il suo server Uniqush, semplicemente da linea di comando

4.3 Architettura

In questo capitolo andrò ad analizzare gli *Use Case Diagram* e l'*Activity Diagram*, ovvero il protocollo, utilizzati per lo sviluppo dell'applicazione. Prima di andare ad analizzare introdurrò cosa è UML (*Unified Modeling Language*) in quanto i primi tre, utilizzano UML.

L'UML è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. Il nucleo del linguaggio fu definito nel 1996 da Grady Booc, Jim Rumbaugh e Ivar Jacobson sotto l'egida dello Object Management Group, consorzio che tuttora gestisce lo standard UML. Il linguaggio nacque con l'intento di unificare approcci precedenti, raccogliendo le migliori prassi nel settore e definendo così uno standard industriale unificato.

L'UML svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione ad oggetti. Gran parte della letteratura di settore usa UML per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile ad un vasto pubblico.

La notazione UML è semi-grafica e semi-formale; un modello UML è costituito da una collezione organizzata di diagrammi correlati, costruiti componendo elementi grafici, elementi testuali formali ed elementi di testo libero. Ha una semantica molto precisa e un grande potere descrittivo.

Il linguaggio è stato progettato con l'obiettivo esplicito di facilitare il supporto software alla costruzione di modelli e l'integrazione di questo supporto con gli ambienti integrati di sviluppo. UML è un linguaggio di modellazione *general purpose*, che fornisce concetti e strumenti applicabili in tutti i contesti.²⁴

Passiamo all'analisi più approfondita dei diagrammi sopra citati e del protocollo utilizzato.

²⁴ Cfr. http://it.wikipedia.org/wiki/Unified_Modeling_Language

4.3.1 *Use Case Diagram*

In UML, gli *Use Case Diagram* (diagrammi dei casi d'uso) sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti ed utilizzati dagli attori che interagiscono col sistema stesso. Sono impiegati soprattutto nel contesto della *Use Case View* (vista dei casi d'uso) di un modello, ed in tal caso si possono considerare come uno strumento di rappresentazione dei requisiti funzionali di un sistema. Tuttavia, non è impossibile ipotizzare l'uso degli UCD in altri contesti; durante la progettazione, per esempio, potrebbero essere usati per modellare i servizi offerti da un determinato modulo o sottosistema ad altri moduli o sottosistemi. In molti modelli di processo software basati su UML, la *Use Case View* e gli *Use Case Diagram* che essa contiene, rappresentano la vista più importante, attorno a cui si sviluppano tutte le altre attività del ciclo di vita del software.

Per capire meglio gli *Use Case Diagram* andiamo ad analizzarli un po' più nel dettaglio. Gli *Use Case Diagram* ha tre *model element* e sono sistema, attori e *use case*.

Il sistema nel suo complesso è rappresentato come un rettangolo vuoto. Questo simbolo viene messo in relazione con gli altri nel senso che i *model element* che rappresentano caratteristiche del sistema, verranno posizionati all'interno del rettangolo mentre quelli che rappresentano entità esterne sono posizionati all'esterno. Questo modello viene molte volte omissso, in quanto la distinzione fra concetti relativi al sistema e concetti relativi al suo contesto si può considerare implicita.

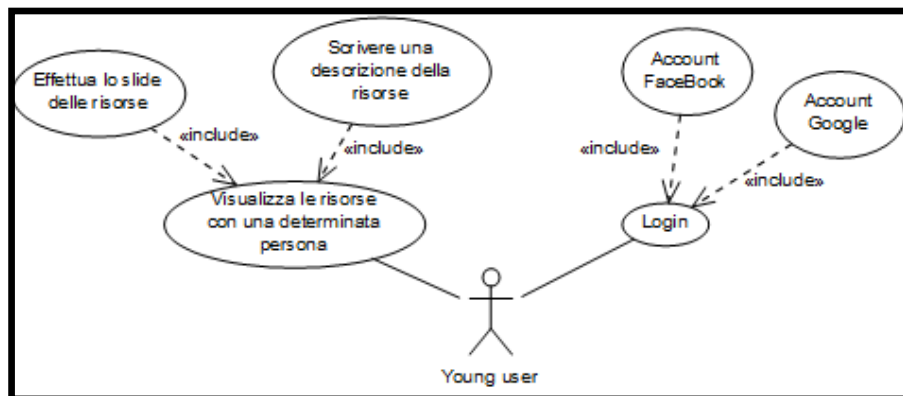
Gli attori sono rappresentati graficamente nel diagramma, da un'icona che rappresenta un uomo stilizzato. Un attore rappresenta un ruolo coperto da un certo insieme di entità interagente col sistema come utenti umani, altri sistemi software, dispositivi hardware, etc.

Uno *use case* è rappresentato graficamente come un'ellisse contenente il nome del caso d'uso, e rappresenta una funzione o servizio offerto dal sistema ad uno o più attori. La funzione deve essere completa e significativa dal punto di vista degli attori che vi partecipano.²⁵

²⁵ Cfr. http://it.wikipedia.org/wiki/Use_Case_Diagram

Ora vado ad analizzare i due *Use Case Diagram* utilizzati per l'applicazione.

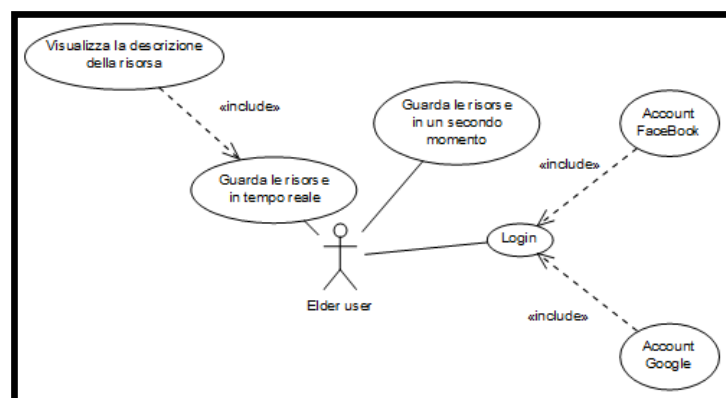
Il primo *Use Case Diagram* riguarda l'utente giovane (*young user*):



Lo *Use Case Diagram*, come detto in precedenza, ci permette di vedere che azioni può intraprendere l'attore in questione. Il nostro utente giovane può effettuare due cose:

- Login tramite un *account* FaceBook oppure un *account* Google;
- Visualizzare le risorse con una determinata persona ovvero iniziare una nuova sessione di sincronizzazione. Mentre condivide le risorse, esso può effettuare lo *slide* di queste ultime in modo da visualizzare pure le restanti risorse oppure scrivere una descrizione della risorsa. Una risorsa può essere una foto come un pezzo di testo.

Il secondo *Use Case Diagram* riguarda l'utente anziano (*elder user*):



L'utente anziano può effettuare tre cose:

- Login, come per l'utente giovane, tramite *account* FaceBook oppure *account* Google;

- Guardare le risorse in tempo reale con l'utente dal quale ha accettato di iniziare la sincronizzazione. Mentre effettua quest'azione esso può visualizzare la descrizione delle risorse scritte dall'utente giovane;
- Guardare le risorse in un secondo momento dopo averle visualizzate in tempo reale con il giovane.

4.3.2 Activity Diagram

L'*activity diagram* è un diagramma definito all'interno dello *Unified Modeling Language* (UML) che definisce le attività da svolgere per realizzare una data funzionalità. Può essere utilizzato durante la progettazione del software per dettagliare un determinato algoritmo. Più in dettaglio, un *activity diagram* definisce una serie di attività o flusso, anche in termini di relazioni tra le attività, i responsabili per le singole attività e i punti di decisione. L'*activity diagram* è spesso usato come modello complementare allo *Use Case Diagram*, per descrivere le dinamiche con cui si sviluppano i diversi *use case*.

Gli *Activity Diagram* sono costruiti da un numero limitato di forme connesse con frecce. Le forme più importanti sono:

- Rettangoli smussati che rappresentano le *Activity*;
- Diamanti che rappresentano le decisioni;
- La barra rappresenta l'inizio (divisione) o la fine (unione) delle attività concorrenti;
- Un cerchio nero rappresenta l'inizio del flusso di lavoro;
- Un cerchio nero circondato rappresenta la fine.

Le frecce indicano in che ordine determinate attività si succedono.

Vado ad analizzare, nel dettaglio, cosa rappresentano le *Activity* e i flussi rappresentati rispettivamente da rettangoli smussati e da frecce.

L'*Activity* rappresenta una specifica attività che deve essere svolta all'interno della funziona. All'interno del rettangolo smussato c'è la descrizione dell'attività.

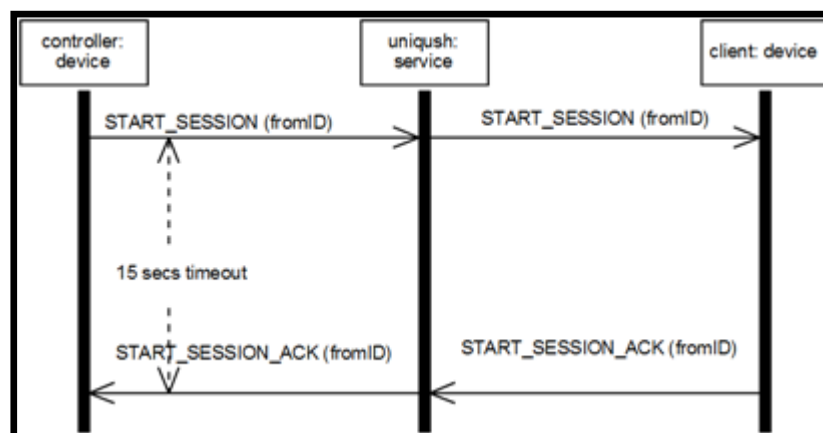
Il flusso è rappresentato da delle frecce orientate, che indicano la sequenza temporale con cui devono essere effettuate le diverse attività. È previsto un simbolo per indicare l'inizio del flusso ed altro per indicarne il termine. Le attività possono essere anche rese in parallelo, in questo caso il punto di divisione è rappresentato da frecce divergenti rispetto al segmento.

Nel caso le attività siano alternative, cioè svolte o meno rispetto ad una scelta, il punto di decisione è rappresentato da dei rombi da cui partono i flussi alternativi.

Il punto di ricongiungimento è reso tramite un segmento su cui le frecce si ricongiungono.²⁶

Gli *activity diagram* utilizzati per sviluppare l'applicazione corrispondono al protocollo implementato attraverso il quale i dispositivi si scambiano i messaggi. Sulla base di queste conoscenze passo ad illustrare i vari *activity diagram* utilizzati per l'applicazione:

1. *Activity diagram* relativo all'inizio della sessione andata a buon fine:

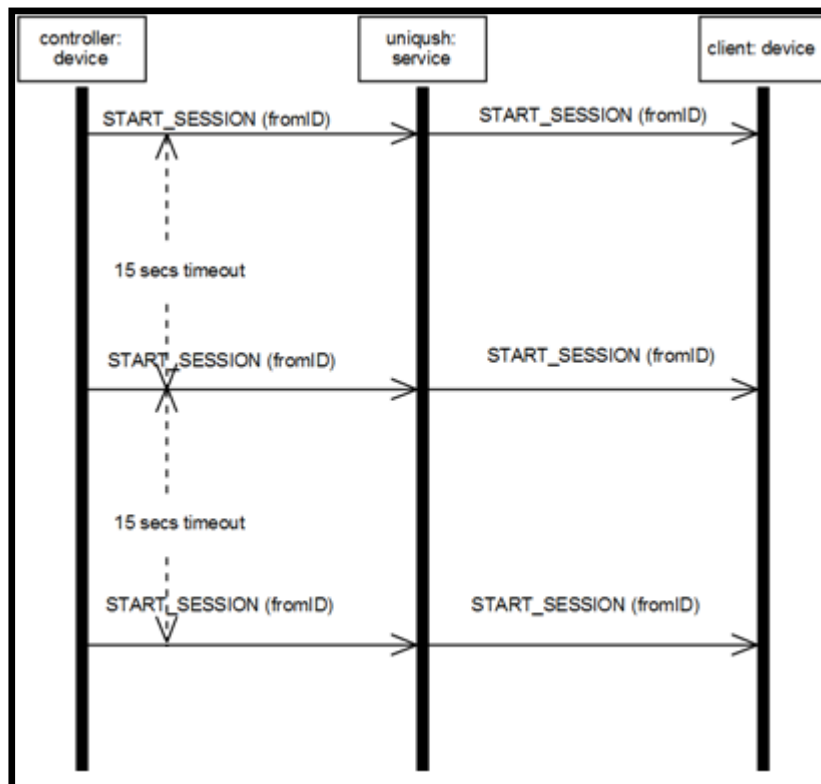


In questo *activity diagram* vediamo lo scambio di messaggi relativo all'inizio della sessione di sincronizzazione andata a buon fine. Il dispositivo che vuole mandare la richiesta di sincronizzazione invia il messaggio **START_SESSION(fromID)** dove il *fromID* è il *topic* a cui appartiene. Una volta inviato il messaggio parte un *timer* per calcolare i 15 secondi che ha a disposizione l'altro dispositivo per rispondere alla richiesta di sincronizzazione. Oltre a far partire il *timer* lo stato del dispositivo passa da **NOT_CONNECTED** a **WAIT_FOR_SESSION_ACK**. Il dispositivo ricevente accetta

²⁶ Cfr. http://it.wikipedia.org/wiki/Activity_diagram

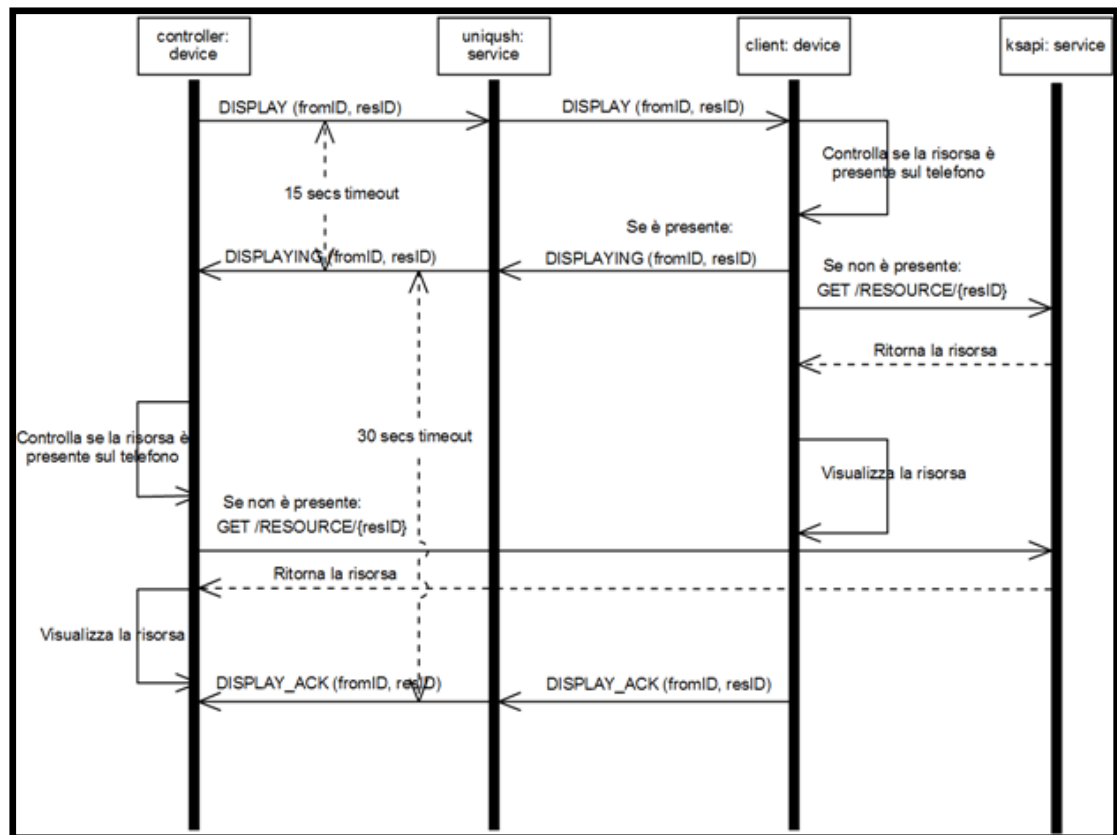
la richiesta di iniziare una sessione di sincronizzazione quindi risponde con un ***START_SESSION_ACK(fromID)*** in modo da far passare il dispositivo chiamante dallo stato ***WAIT_FOR_SESSION_ACK*** a ***READY*** ed il dispositivo ricevente passa da ***NOT_CONNECTED*** a ***READY***.

2. *Activity diagram* relativo all'inizio di una sessione non andata a buon fine:



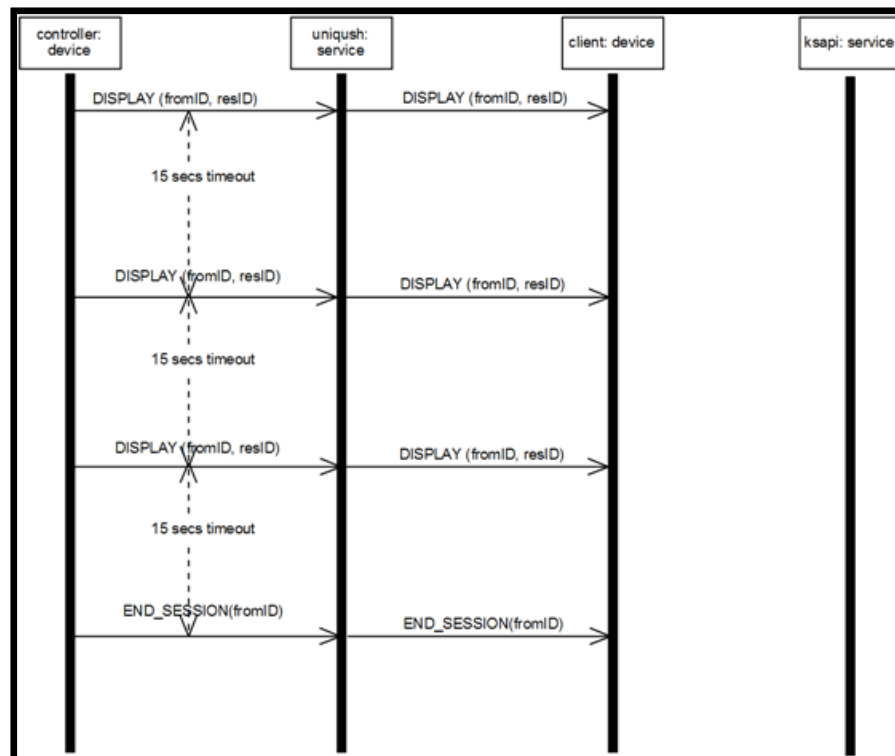
In questo *activity diagram* vediamo lo scambio di messaggi relativo all'inizio della sessione di sincronizzazione non andata a buon fine. Il dispositivo chiamante manda il messaggio ***START_SESSION(fromID)*** al dispositivo con cui vuole inviare la sincronizzazione. Se il dispositivo ricevente entro 15 secondi, non invia un ***START_SESSION_ACK(fromID)*** il dispositivo chiamante rinvia la richiesta di iniziare una sessione per altre 2 volte. Se il dispositivo ricevente continua a non rispondere, il dispositivo chiamante smette di inviare le richieste per iniziare la sessione e passa dallo stato ***WAIT_FOR_SESSION_ACK*** allo stato di ***NOT_CONNECTED***.

3. *Activity diagram* relativo alla visualizzazione di una determinata risorsa andata a buon fine:



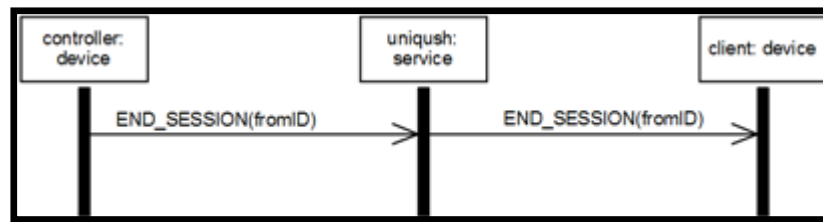
In questo *activity diagram* vediamo lo scambio di messaggi relativo alla visualizzazione di una determinata risorsa andata a buon fine. Il dispositivo chiamante invia il messaggio **DISPLAY(fromID, resID)** dove con *resID* si indica l'url, o l'identificativo, della risorsa mentre con *fromID* s'intende a chi è destinato quel messaggio. Al momento dell'invio del messaggio parte un *timer* di 15 secondi entro il quale deve essere ricevuto il messaggio e lo stato del dispositivo passa da **READY** a **WAIT_FOR_DISPLAY_ACK**. Il dispositivo ricevente controlla se la risorsa è disponibile localmente. Se è così invia subito i due messaggi **DISPLAYING(fromID, resID)** e subito dopo il messaggio **DISPLAY_ACK(fromID, resID)** facendo passare così il dispositivo chiamante da **WAIT_FOR_DISPLAY_ACK** a **READY**. Nell'eventualità la risorsa non sia disponibile sul dispositivo, quest'ultimo prima invierà il messaggio **DISPLAYING(fromID, resID)** facendo così partire un timer dalla durata di 30 secondi, poi contatterà le API del programma usando il comando **GET /RESOURCE/{resID}** prelevando su la risorsa. Una volta prelevata la risorsa esso invierà il messaggio **DISPLAY_ACK(fromID, resID)** facendo passare così il dispositivo chiamante da **WAIT_FOR_DISPLAY_ACK** a **READY**.

4. *Activity diagram* relativo alla visualizzazione di una determinata risorsa non andata a buon fine:



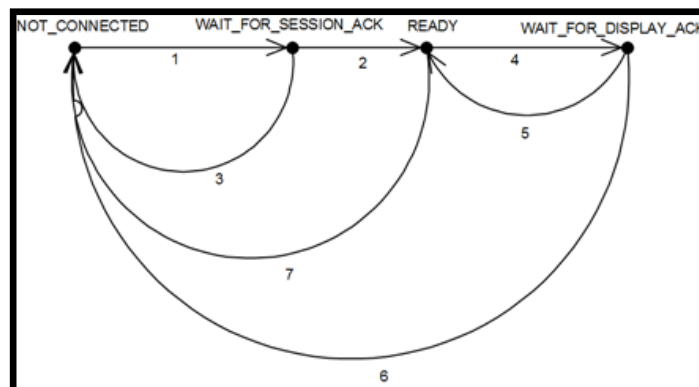
In questo *activity diagram* vediamo lo scambio di messaggi relativo alla visualizzazione di una determinata risorsa non andata a buon fine. Il dispositivo chiamante invia il messaggio **DISPLAY(fromID, resID)** dove con *resID* si indica l'url, o l'identificativo, della risorsa mentre con *fromID* s'intende a chi è destinato quel messaggio. Al momento dell'invio del messaggio parte un *timer* di 15 secondi entro il quale deve essere ricevuto il messaggio e lo stato del dispositivo passa da **READY** a **WAIT_FOR_DISPLAY_ACK**. Se il dispositivo chiamante non riceve il messaggio **DISPLAYING(fromID, resID)** esso continua ad inviare il messaggio **DISPLAY(fromID, resID)** per altre due volte con un intervallo di 15 secondi l'uno dall'altro al termine del quale invierà il messaggio **END_SESSION(fromID)** per terminare la sessione di sincronizzazione e passando così dallo stato **WAIT_FOR_DISPLAY_ACK** a **NOT_CONNECTED**. Nell'eventualità il dispositivo ricevente inviasse solo il messaggio **DISPLAYING(fromID, resID)**, il dispositivo chiamante invierebbe il messaggio **DISPLAY(fromID, resID)** con un intervallo di 30 secondi l'uno dall'altro. Inviato l'ultimo **DISPLAY(fromID, resID)** il dispositivo chiamante invierà il messaggio **END_SESSION(fromID)** ed il suo stato passerà da **WAIT_FOR_DISPLAY_ACK** a **NOT_CONNECTED**.

5. *Activity diagram* relativo alla fine di una sessione:



In questo *activity diagram* vediamo lo scambio di messaggi relativo alla terminazione di una sessione di sincronizzazione. Il dispositivo che decide di terminare la sessione di sincronizzazione, invia il messaggio **END_SESSION(fromID)** e passa da qualunque sia il suo stato attuale allo stato di **NOT_CONNECTED** questo per poter permettere al dispositivo di iniziare un'altra sessione di sincronizzazione con un altro dispositivo. Colui che riceve il messaggio ritorna alla schermata principale e il suo stato passa a **NOT_CONNECTED**.

6. Riassunto degli stati in cui si può trovare il dispositivo chiamante:



In quest'immagine vediamo quando e quali sono gli stati in cui si può ritrovare un dispositivo. All'inizio vediamo il passaggio del dispositivo dallo stato **NOT_CONNECTED** allo stato **WAIT_FOR_SESSION_ACK** (1). Questo lo si può trovare quando si vuole iniziare una sessione di sincronizzazione ovvero quando il dispositivo chiamante ha inviato il messaggio **START_SESSION(fromID)**. Nell'eventualità in cui il dispositivo ricevente accettasse l'invito, il dispositivo chiamante passerebbe allo stato di **READY** (2). Se il dispositivo ricevente rifiutasse l'invito, il dispositivo chiamante ritornerebbe allo stato di **NOT_CONNECTED** (3). Dallo stato di **READY** si passa subito allo stato di **WAIT_FOR_DISPLAY_ACK** (4). Questo passaggio è possibile solo dopo aver stabilito la connessione in quanto il dispositivo chiamante, invierebbe il messaggio **DISPLAY(fromID, resID)** ed

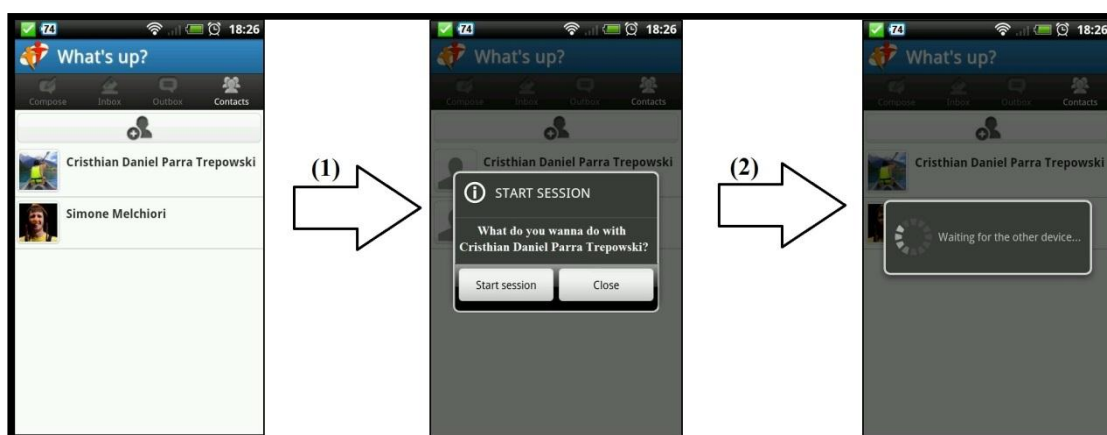
attenderebbe un **DISPLAY_ACK**(*fromID*, *resID*) da parte del dispositivo ricevente. Nell'eventualità il dispositivo ricevente inviasse il messaggio, il dispositivo chiamante passerebbe dallo stato **WAIT_FOR_DISPLAY_ACK** allo stato **READY** (5). Se il dispositivo ricevente non gli inviasse il messaggio, il dispositivo chiamante invierebbe il messaggio **END_SESSION**(*fromID*) e passerebbe dallo stato **WAIT_FOR_DISPLAY_ACK** allo stato **NOT_CONNECTED** (6) pronto, così, per iniziare un'altra sessione di sincronizzazione. Dallo stato **READY** il dispositivo chiamante, ed anche ricevente, può passare allo stato di **NOT_CONNECTED** (7) inviando o ricevendo il messaggio **END_SESSION**(*fromID*).

Il dispositivo ricevente ha solamente gli stati di **NOT_CONNECTED** e di **READY** in quanto non deve attendere nessun messaggio di conferma da parte del dispositivo chiamante.

4.4 Immagini dell'applicazione sviluppata

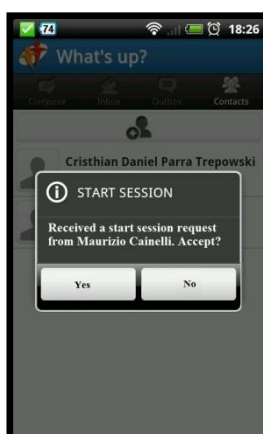
In questo sottocapitolo andrò ad illustrare ed a mostrare le immagini relative all'applicazione sviluppata. Analizzerò ciò che succede prima nella parte del dispositivo chiamante e subito dopo dalla parte del dispositivo ricevente.

La prima immagine mostra cosa succede quando si vuole iniziare una sessione di sincronizzazione con una determinata persona:

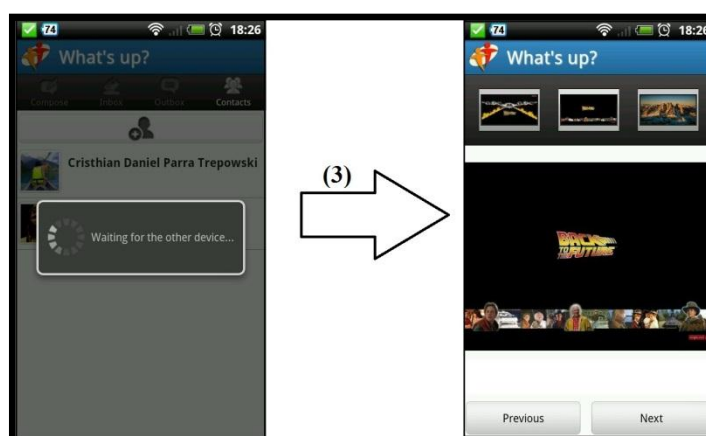


Cliccando sul contatto con il quale si vuole avviare una sessione appare un messaggio di conferma e cliccando su 'Start session' (1) il dispositivo invierà il messaggio al server Uniqush che lo inoltrerà al dispositivo corretto. In attesa del messaggio di risposta il dispositivo mostrerà una schermata di caricamento (2).

Il dispositivo ricevente avrà invece, la seguente schermata:

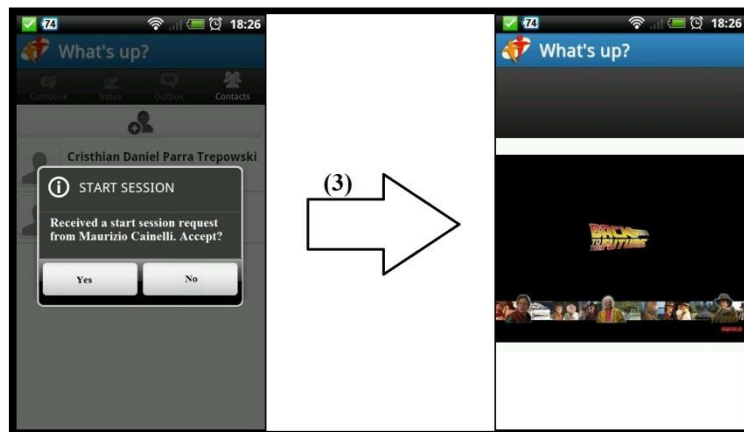


Così facendo il dispositivo ricevente ha la possibilità di accettare o meno la richiesta di sincronizzazione. Premendo su 'Yes' il dispositivo invierà il messaggio ***START_SESSION_ACK(fromID)*** mentre, premendo 'No' rifiuterà la richiesta del dispositivo. Nell'eventualità che il dispositivo ricevente accettasse la richiesta di sincronizzazione, questo è ciò che succederebbe dalla parte del dispositivo chiamante:



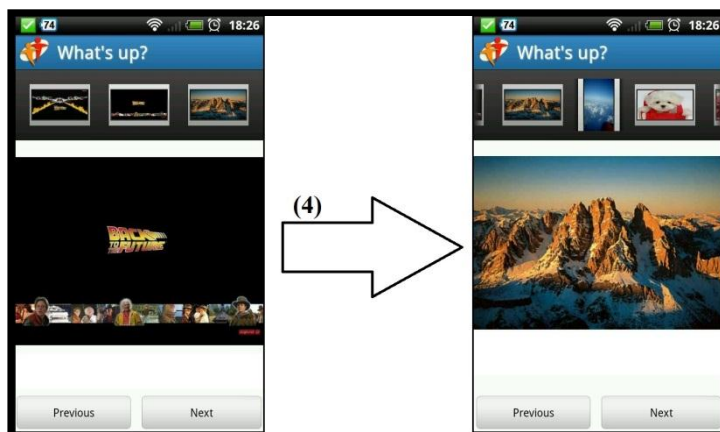
Il dispositivo chiamante invierebbe il messaggio ***DISPLAY(fromID, resID)*** per poter essere sincronizzato con il dispositivo ricevente (3).

Dall'altra parte si avrà una situazione di questo tipo:

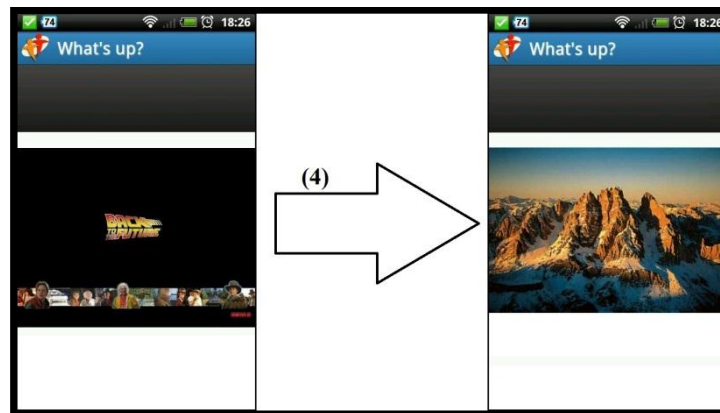


Una volta ricevuto il messaggio *DISPLAY(fromID, resID)*, il dispositivo ricevente invierà il messaggio *DISPLAYING(fromID, resID)* ed una volta visualizzata la risorsa, potrà inviare *DISPLAY_ACK(fromID, resID)* (3).

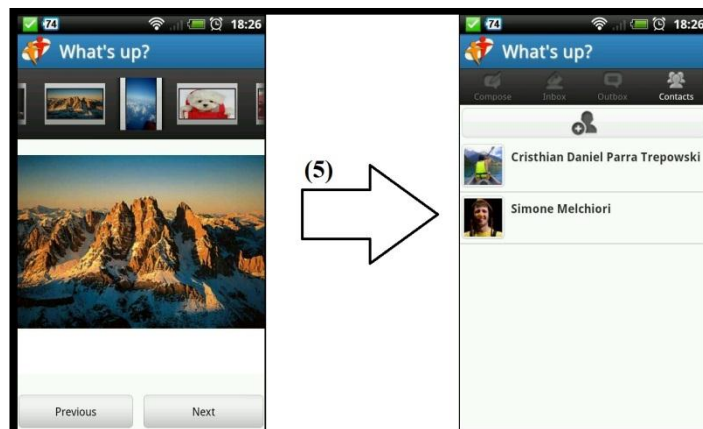
Il dispositivo ricevente non ha la possibilità di vedere tutte le altre risorse sotto forma di galleria, e non ha nemmeno i tasti 'Previous' e 'Next' in quanto lui non può cambiare le immagini. Nell'eventualità il dispositivo chiamante voglia cambiare immagine esso dovrà inviare il messaggio *DISPLAY(fromID, resID)* con il *resID* differente da quello dato in precedenza (4). Così facendo, si effettuerà un cambio dell'immagine ed un relativo scambio di messaggi. Questo è ciò che apparirà dalla parte del dispositivo chiamante:



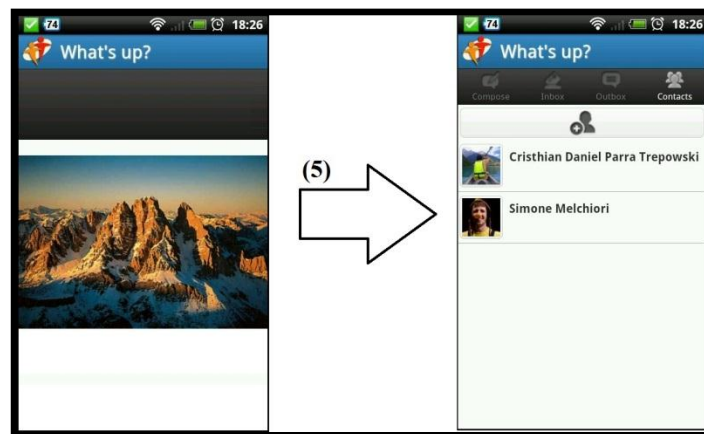
Mentre questo è ciò che si vedrà dal dispositivo ricevente dove con (4) indico l'invio dei messaggi **DISPLAYING** alla ricezione di **DISPLAY** e **DISPLAY_ACK** alla visualizzazione della risorsa richiesta:



Il dispositivo chiamante e ricevente potranno terminare, ogni qualvolta vogliano, la loro sincronizzazione premendo il tasto indietro sul loro telefono oppure premendo l'opzione presente nel menù e questo comporterà l'invio del messaggio **END_SESSION(fromID)** (5) portando i dispositivi dallo stato di **READY** allo stato di **NOT_CONNECTED**. Una volta premuto il tasto questa sarà ciò che succederà per il dispositivo chiamante:



Questo è ciò che succederà al dispositivo ricevente:



Se si riceve o si invia, il messaggio ***END_SESSION(fromID)*** esso riporterà il
dispositivo alla lista dei contatti.

Capitolo 5

Risultati e conclusioni

In questo capitolo parlerò dei risultati ovvero andrò a spiegare la procedura che si effettua per poter mettere l'applicazione su *Google Play*, dirò a che *range* di utenti è dedicata quest'applicazione e trarrò delle conclusioni, ovvero dirò le novità che si vogliono introdurre su quest'applicazione e se essa è funzionale o meno.

5.1 Risultati

Come detto in precedenza, in questo sottocapitolo andrò a spiegare come si fa a mettere l'applicazione su *Google Play* ed a che *range* di utenti è rivolta l'applicazione da me sviluppata. Iniziamo con la procedura da effettuare per poter pubblicare applicazioni su *Google Play*. Prima di tutto bisogna registrarsi e pagare una quota di iscrizione di 25 dollari. Google impone questa quota di iscrizione in quanto vuole prevenire la possibilità di spam da parte di utenti. Una volta registrati per caricare le applicazioni, si va sulla console degli sviluppatori e si seleziona 'Carica applicazioni'. Verrà chiesto di inserire determinate informazioni relative all'applicazione che si vuole pubblicare divise in quattro categorie che sono:

- Carica risorse;
- Dettagli di inserzione;
- Opzioni di pubblicazione;
- Informazioni di contatto.

Vado ad analizzare ciò che viene richiesto da ognuna di queste categorie iniziando da 'Carica risorse':

- Dimensioni file APK: le dimensioni massime supportate sono pari a 50 MB;
- File .apk applicazione bozza: quando carichi un file .apk, puoi salvarlo come bozza mentre modifichi gli altri aspetti dell'elenco. I nomi dei pacchetti dei file delle applicazioni sono univoci e definitivi. Non è possibile eliminare o riutilizzare in futuro i nomi dei pacchetti;

- *Screenshot*: sono obbligatori due *screenshot* per mostrare l'applicazione. Gli altri sei sono facoltativi;
- Icona applicazione ad alta risoluzione: obbligatoria;
- Immagine promozionale: facoltativa;
- Immagine della funzione: facoltativa;
- Video promozionale: facoltativa.

Per quanto riguarda la categoria 'Dettagli di inserzione', questo è ciò che viene richiesto:

- *Lingua*: quest'opzione consente di indicare la lingua dell'applicazione. La lingua predefinita è l'inglese americano. Saranno disponibili altre lingue non appena diventeranno disponibili i dispositivi *Android* in tali lingue;
- *Titolo*: il nome dell'applicazione così come viene visualizzato in *Google Play*. Si può aggiungerne uno per lingua;
- *Descrizione*: la descrizione visibile dell'applicazione in *Google Play*. Si può usare la funzione di traduzione per tradurre la descrizione in altre lingue. Questo campo ha un limite di 4000 caratteri;
- *Modifiche recenti*: la funzione delle modifiche recenti ti consente di aggiungere note riguardo alle specifiche modifiche apportate alla versione più recente dell'applicazione. Si può usare questo campo per informare gli utenti delle modifiche apportate all'applicazione;
- *Testo promozionale*: testo che accompagna l'immagine promozionale in posizioni di primo piano in *Google Play*;
- *Tipo di applicazione*: *Google Play* è suddiviso in due tipi di applicazioni principali e sono 'Applicazioni' e 'Giochi';
- *Categoria*: Bisogna selezionare una categoria per l'applicazione.

Ora andrò ad analizzare la terza categoria ovvero 'Opzioni di pubblicazione':

- *Protezione della copia*: la protezione della copia potrebbe impedire la copia delle applicazioni da un dispositivo. Essa aumenta la quantità di memoria del cellulare necessaria per l'installazione dell'applicazione. A breve Google abbandonerà questa funzione. Google consiglia di implementare il servizio di licenze di *Google Play* se l'applicazione è a

pagamento altrimenti di implementare il proprio schema di protezione dalla copia.

- Località: si tratta delle località in cui puoi distribuire le tue applicazioni in quanto:
 - non in tutte le località elencate dalla lista, esistono attualmente utenti con dispositivi *Android*;
 - puoi selezionare le singole località o selezionare l'opzione 'Tutte le località attuali e future'. Quest'opzione indica che, non appena Google aggiungerà altre località, tali località saranno attivate per l'applicazione.

L'ultima categoria è 'Informazioni di contatto':

- Bisogna selezionare un canale di supporto per l'applicazione, vale a dire sito web, e-mail o telefono;
- Queste informazioni sono visualizzabili dagli utenti su *Google Play*;
- Si può scegliere di offrire più canali per l'assistenza.²⁷

Una volta compilati questi campi, la nostra applicazione ha il diritto ad essere pubblicata su *Google Play* e quindi ad essere scaricata dagli utenti. Passiamo ora all'analisi del *range* di utenti relativo all'applicazione sviluppata.

L'applicazione sviluppata, ha come *range* di utenti tutte le categorie partendo quindi dalle persone più anziane fino ad arrivare a quelle più giovani. Questo perché l'applicazione può essere utilizzata da tutti per i seguenti motivi:

- Interfaccia *user-friendly* ovvero non troppo complicata da capire;
- Facile da usare;
- Colui che riceve la chiamata, di solito l'anziano, non deve effettuare nessuna azione dopo aver accettato la sincronizzazione e quindi eventuali tremolii della mano non creano problemi. Il ricevente deve semplicemente guardare le immagini passare sul suo telefono/tablet;
- Avvicina le persone meno giovani al mondo tecnologico;

²⁷ Cfr. <http://support.google.com/googleplay/android-developer/bin/answer.py?hl=it&answer=113469&topic=2365624&ctx=topic>

- L'utente anziano può iniziare la sincronizzazione in modo da poter condividere le proprie immagini con l'utente più giovane.

5.2 Conclusioni

L'applicazione sviluppata è stata integrata nell'applicazione principale *What's Up?*, è funzionale, funziona a dovere ed è scaricabile da *Google Play* (per trovare più velocemente l'applicazione, cercare LifeParticipation).

Ciò che si vuole introdurre in futuro riguardante l'applicazione è:

- Avviando la sincronizzazione si vuole avviare anche una chiamata vocale con la persona desiderata;
- Implementare la possibilità di taggare i propri contatti sulle foto;
- Implementare la possibilità di fare nuove foto durante la sincronizzazione, condividerle con la persona con cui ho avviato la sincronizzazione ed iniziare a sincronizzarle subito;

BIBLIOGRAFIA

<http://it.wikipedia.org/wiki/>

<http://en.wikipedia.org/wiki/>

<http://wearesocial.it/blog/>

<http://www.lifeparticipation.org/>

<http://mqtt.org/>

<https://developers.google.com/>

<http://developer.android.com/>

<http://uniqush.org/>

<http://support.google.com/>