Alma Mater Studiorum · Università di Bologna

**FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI**
**Corso di Laurea in Informatica per il management**

# APostrophe
# WiFi Sharing

**Relatore:**
**Chiar.mo Prof.**
**Luciano Bononi**

**Correlatore:**
**Dott. Luca Bedogni**

**Presentata da:**
**Vincenzo Morelli**

**II Sessione**
**2016/2017**

# ABSTRACT

Negli ultimi anni il tema dell'economia della condivisione ha assunto un piano sempre più attuale, lo sviluppo delle tecnologie mobile, contemporaneamente, è in continua ed esponenziale crescita, le applicazioni necessitano sempre più di un accesso alla rete internet sicuro ed efficiente.

Questa tesi è incentrata sullo sviluppo di un'applicazione per sistemi Android che consenta la creazione di un network di utenti e permetta loro di condividere il proprio router casalingo.

I dispositivi condivisi saranno collocati sulla mappa tramite un algoritmo di posizionamento che abbina le meccaniche di geolocalizzazione a quelle del WiFi Manager presente sui dispositivi.

Selezionando l'hotspot più vicino sarà quindi possibile richiedere la connessione per un periodo limitato di tempo così da avere accesso ai servizi Facebook, Instagram, Maps e YouTube.

Per la progettazione e l'implementazione di Apostrofe sono stati affrontati e approfonditi argomenti riguardanti: l'implementazione di un algoritmo di posizionamento , l'implementazione di un web service , cifratura e trasmissione dati tramite protocollo HTTP .

# Indice

# Elenco delle figure

# Listings

CAPITOLO 1

# *Introduzione*

# 1 Introduzione

Il mercato degli smartphone non è ancora saturo. La produzione e la vendita di dispositivi mobili continua a crescere a dismisura tanto che ad oggi, come affermato dal mobility report di Ericsson, nel mondo vi sono più sim che esseri umani.
Sempre più lontano da un convenzionale cellulare e sempre più vicino ad un inusuale computer, lo smartphone costituisce un importante ponte tra l'utente e la rete, gestendo per la quasi totalità quello che è considerato sotto certi aspetti il trampolino di lancio verso una "società 2.0" : i social network.
La richiesta di una connessione stabile che permetta di interagire con questo mondo è un problema concreto che i gestori del traffico mobile non sempre riescono a soddisfare nella sua totalità. Molto spesso, infatti, la connettività offerta con le tariffe telefoniche non è soddisfacente per problemi di ricezione, oppure più semplicemente, non è sufficiente per quello che è l'utilizzo dell'utente medio.
Per far fronte a questa necessità, anche gli enti pubblici si stanno attivando per la creazione di aree provviste di accesso ad internet illimitato, tuttavia, specie in Italia, il servizio appare ancora lento, malfunzionante e poco diffuso.

*E se gli utenti condividessero la propria connessione casalinga?*

Si immagini di poter girare per le strade della propria città, o magari in un paese di campagna di uno stato estero ed essere sempre connessi ad una rete veloce, stabile e senza limiti di dati scambiabili, dove , tramite il pagamento di una piccola tassa oppure addirittura gratuitamente, si possa accedere a tutti i router della zona in totale sicurezza e affidabilità per poter inviare una mail, consultare le mappe o più semplicemente per condividere una foto su un social network. Questo è fulcro del progetto di tesi.
Apostrofe è un'applicazione per dispositivi mobili che offre un servizio di registrazione, condivisione ed utilizzo di qualsiasi punto di accesso alla rete internet, ogni persona dotata di uno smartphone può installare l'applica-

zione e cominciare a connettersi.
Se un gran numero di utenti decidessero di aderire a questo progetto, si assisterebbe alla nascita di una vera e propria community di gente che non solo sfrutta i servizi, ma ne propone di nuovi a sua volta, creando un circolo virtuoso che garantirebbe all'app un'ampia diffusione.

## 1.1 Stato dell'arte

In questa sezione sono descritti i progetti affini ad Apostrofe e delineati punti di forza e debolezza che questa tesi si promette di affrontare.

### 1.1.1 WiFi community

Negli ultimi quindici anni, abbiamo assistito a numerosi tentativi di offerta della connessione ad internet attraverso differenti tecnologie e modelli di business. Sorprendentemente, questi tentativi non partivano dagli operatori specializzati nell'offerta di una connessione 3G, ma da individui organizzati in wireless communities che provvedevano ad offrire agli altri membri un accesso gratuito alla rete, utilizzando il proprio access point privato. Al giorno d'oggi, per diverse ragioni le reti 3G e 4G sono di gran lunga le soluzioni preferite dagli utenti, la più importante è la facilità e il basso impiego di mezzi cognitivi tramite il quale si può usufruire del servizio.
La chiave per il successo di una community, diversamente da una tecnologia offerta da un privato, è il coinvolgimento delle masse nel condividere le risorse per partecipare attivamente alla vita del gruppo stesso.
Risulta importante quindi capire perchè, e in che circostanze, gli utenti potrebbero partecipare attivamente ed essere attratti dai servizi che la community offre.
Numerose ricerche hanno infatti dimostrato, che il punto fondamentale attorno al quale ruota il successo delle community è **l'incentivo**, più è grande il beneficio ottenibile partecipando e più l'utente è invogliato a partecipare.
Possiamo distinguere le community in due categorie: pure e ibride.[1] Le pure sono communities interamente costruite ed utilizzate da membri che condividono il loro access point con gli altri utenti in maniera indipendente

ed organizzata, ma soprattutto gratuita.
Le ibride, invece, sono supportate da membri che condividono il proprio access point offrendo soluzioni tecniche migliori e vari incentivi in cambio del diritto di utilizzo della connessione. Fonte di guadagno sono le partnership offerte agli utilizzatori oppure le publicità all'interno dei servizi.
Apostrofe è considerabile a tutti gli effetti della tipologia ibrida e punta a sfruttare l'attrattività di un possibile , seppur non consistente guadagno, per permettere alla community una rapida crescita ed una rapida diffusione del servizio.

### 1.1.2 Il progetto FON

Prima di Apostrofe, molti sono stati i tentativi di realizzazione di un network di utenti che potesse condividere la propria rete domestica al fine di creare una rete globale, tuttavia, specie in Italia, non hanno riscontrato grande successo se non con gli utenti più esperti ed informati circa la tematica delle *Wireless community networks.*
Il progetto più attuale e largamente diffuso è senza dubbio quello portato avanti dall'organizzazione spagnola "Fon" [2] , fondata nel 2006 da Martin Varsavsky ed attualmente finanziata da colossi dell'informatica del calibro di Microsoft e Google.
Fon nasce con il preciso scopo di garantire a tutti i membri aderenti al progetto un accesso ad internet sicuro ed efficiente: al giorno d'oggi, i dispositivi connessi al network Fon risultano più di venti milioni .
Per poter usufruire dei servizi Fon, in principio, era necessario l'acquisto di un access point da installare parallelamente al router casalingo , al fine di poter creare una rete "WiFi Guest" , a cui ogni utente potesse connettersi.
Durante i primi mesi dal rilascio del progetto, gli utenti aderenti erano più di diecimila, nel 2008 Fon dichiara di aver toccato il milione di "foneros" .
Dapprima movimento nato come associazione, ha mutato poi leggermente il suo schema, differenziandosi completamente dalle altre community wireless diffuse nel mondo. Sono stati introdotti infatti vari livelli di affiliazione [3] che tuttavia dal 2015 non sono più utilizzati:

Linus: è il membro no profit del movimento Fon. Un Linus acquista un

router direttamente da Fon, oppure scarica il firmware dal sito dell'associazione ed entra a far parte della comunità.In cambio dell'utilizzo dei dispositivi messi a disposizione da tutti gli altri membri della community, anch'egli presta il proprio router al servizio destinando parte della connessione agli altri foneros.

Bill: tutti coloro che scelgono di adottare le soluzioni FON. Da giugno 2007 possono connettersi a tutti gli hotspot della rete FON, al pari dei Linus, ma percepiranno il 50% dei profitti generati dagli utenti Alien che si collegheranno al loro hotspot.

Alien: chiunque può registrarsi ad un hotspot Fon. Se non si è Linus, infatti, basta acquistare un pass che permetta all'utente di navigare per un tempo determinato. In alternativa, con l'introduzione del WiFiAds è possibile navigare 15 min visionando un breve spot pubblicitario. Tuttavia, questa connessione si dimostra anonima e non tracciabile esponendo chi condivide la connessione a potenziali rischi legali.

Dopo il 2015 tali ruoli sono decaduti per lasciare spazio a metodiche sempre più semplici.
La nuova modalità di partecipazione alla rete Fon consiste nell'acquistare direttamente un router Fon presso la casa distributrice, e questo dà accesso alla rete Fon e a sua volta permette agli altri utenti di utilizzare la propria rete di casa come access point, sempre tramite condivisione per mezzo di rete pubblica con solo accesso a internet.
Attraverso partnership strategiche con le più grandi compagnie telefoniche di tutto il mondo, è adesso possibile abbonarsi a partner di Fon, nel caso dell'Italia a Vodafone, e questo concede non solo l'accesso alla rete Fon per tutta la durata dell'abbonamento ma rende il proprio router Vodafone nodo di connettività della network community.

### 1.1.3 La community WiMan

Fondata nel 2012 da due ragazzi pugliesi, WiMan[4] è una startup tutta italiana nata con lo scopo di semplificare l'accesso al WiFi e soddisfare il bisogno di WiFi gratuito.
Oggi giorno si occupa di connettività globale e mira alla mappatura di tutte le reti pubbliche presenti con l'obbiettivo di avere sempre un accesso garantito alla rete internet.
A partire dal 2015 , anno di lancio dell'applicazione su dispositivi android , WiMan è cresciuta in maniera esponenziale contando attualmente sessanta milioni di dispositivi connessi alla rete.

Inizialmente pensata come sistema per la mappatura di WiFi pubblici e per la condivisione della rete per attività commerciali adesso consente anche la condivisione di dispositivi privati utilizzabili con e senza autenticazione.
L'applicazione di per se risulta molto ben strutturata : è provvista di un menu che consente di gestire i dispositivi collegati al proprio account, una mappa della posizione dei WiFi registrati , un pannello per la gestione del proprio profilo e la possibilità di scaricare la mappatura delle reti anche offline per l'utilizzo dei servizi in totale assenza di connessione .
Vera novità , tuttavia , è l'implementazione di un nuovo WiFi Manager che consente di collegarsi e visualizzare le diverse reti direttamente dall'applicazione fornendo anche la distanza dell'utente dal router per ottenere una connessione ottimale.

Al lancio dell'applicazione viene richiesto di autenticarsi tramite social network o registrazione per l'identificazione dell'utente che andrà ad usufruire del servizio , si accede quindi alla schermata del WiFi Manager che presenta subito i dispositivi più vicini a cui connettersi . Questi ultimi possono essere di tre tipologie differenti :

WiFi verificato: Almeno un utente ha utilizzato tale WiFi con successo.

Richiede login: WiFi pubblico che richiede autenticazione per essere utilizzato.

Non verificato: Nessun utente WiMan ha usato tale WiFi finora.

A seconda delle esigenze dell'utente, è quindi possibile scegliere il router a cui connettersi automaticamente senza bisogno di inserire password, gestite interamente dal sistema . Essere connessi dà anche la possibilità di testare la connessione tramite uno speedtest, disponibile all'interno dell'app , e permette di valutare la qualità del servizio come feedback per gli altri utenti.
Qualora si volesse entrare attivamente a far parte della community WiMan è possibile registrare il proprio router casalingo nei database dell'applicazione direttamente dal proprio dispositivo , è sufficiente selezionarlo nel pannello di aggiunta ed inserire la password correlata . Automaticamente verrà verificato e reso disponibile a tutti gli utenti WiMan. Le informazioni sul dispositivo come password ed SSID sono modificabili in qualsiasi momento alla voce del menu "i miei WiFi" , tramite la quale è anche possibile controllare il numero di accessi ed il numero di byte di dati scambiati da utenti che hanno usufruito della connessione.

Il profilo utente è anch'esso molto dettagliato : dispone, infatti, di una schermata di descrizione delle attività svolte , offre la possibilità di controllare la cronologia di tutte le operazioni svolte all'interno dell'app, fornisce interessanti dettagli sugli utenti attivi e sul loro livello di partecipazione alla vita della community. Per quel che concerne la prima schermata, vengono visualizzati oltre al grafico dei propri dati scambiati , anche le statistiche d'uso del WiFi, come il numero di connessioni ed il numero di reti diverse a cui ci si è connessi.
Nel profilo utente, particolare rilevanza è data dalla presenza di un sistema "meritocratico" per la partecipazione alla vita della community: effettuare speedtest , registrare dispositivi oppure recensire connessioni messe a disposizione da altri membri permette di ottenere veri e propri "riconoscimenti" che aumentano il livello di affidabilità dell'utente agli occhi della community ed allo stesso tempo lo spingono a contribuire sempre più al miglioramento dei servizi.

Da pochi mesi è stato anche sviluppato un software *SDK Free WiFi mobile* per permettere agli sviluppatori delle applicazioni di terze parti di integrare la funzione dell' *auto-connect* al miglior Free WiFi in zona, nelle proprie app, portando ulteriori benefici ai loro utenti.

Altro punto di forza della startup bolognese è la fornitura di router WiMan . In tutto e per tutto simile a quanto offerto da Fon, il router WiMan , collegato al router della propria attività commerciale, ma anche ad un router casalingo, crea una rete distinta e parallela che permettete alla clientela di connettersi ad internet dosando l'utilizzo della banda in caso di troppe connessioni e garantendo massima sicurezza, poiché gli utenti si connetteranno alla seconda rete creata e non sarà necessario fornire le credenziali di quella privata. Adoperando questo router si possono monitorare connessioni , capire quali utenti utilizzano il servizio e conseguentemente proporre loro soluzioni pubblicitarie altamente personalizzate.

### 1.1.4 WI FI ITALIA IT

Infine, è utile analizzare il tema del WiFi sharing dal punto di vista degli enti pubblici per delineare quelli che sono i passi avanti che lo stato italiano ha svolto per abbracciare questo mondo sempre più attuale.
Il Progetto WiFi Italia It [5]ha come obiettivo principale quello di permettere a cittadini e turisti, italiani e stranieri, di connettersi gratuitamente e in modo semplice a una rete WiFi libera e diffusa su tutto il territorio nazionale. A tale scopo è stata progettata una APP che permetterà di usufruire della rete per tutti i nodi fondamentali che l'iniziativa statale metterà a disposizione.
Per fare ciò, il Ministero dello Sviluppo Economico ha dato il via ad un processo che consentirà di:

- Federare tutte le reti pubbliche e private per avere un unico e semplice sistema di accesso al Wi-Fi a disposizione di cittadini e turisti;
- Far crescere la rete di accessi Wi-Fi, in particolare nei luoghi del turismo e della cultura;
- Sfruttare la rete di accessi per arricchire di dati l'ecosistema del turismo (analisi statistica dei dati, opportunamente anonimizzati, per studiare comportamenti e preferenze degli utenti e conseguentemente migliorare i servizi).

*Pietra d'angolo del progetto WiFi Italia It è il Protocollo per la diffusione di piattaforme intelligenti al servizio del turista sul territorio italiano siglato dal Ministero dello Sviluppo Economico, dal Ministero dei Beni e le Attività Culturali e dall'Agenzia per l'Italia Digitale. Il protocollo promuove, attiva e sviluppa processi d'innovazione volti alla creazione di un ecosistema digitale del turismo in grado, da un lato, di facilitare l'accesso dei cittadini e dei visitatori al patrimonio artistico, naturale e culturale distribuito sull'intero territorio nazionale, dall'altro di creare un ambiente fertile per il settore privato all'interno del quale sviluppare applicativi e servizi a valore aggiunto.*
Nelle versioni future della APP, sarà possibile la registrazione con SPID e

l'accesso per i PC portatili e pad. L'applicazione non consentirà solo la connessione, i suoi futuri sviluppi mirano anche ad offrire una serie di servizi messi a disposizione da enti ed amministrazioni apposta per chi sceglierà WI FI Italia.
Può essere installata sul proprio dispositivo mobile e, una volta compiuta la registrazione, permette di gestire il collegamento e l'autenticazione sulla rete WiFi sotto la cui copertura ci si trova in quel momento. Una volta scaricata la App basterà seguire la procedura di registrazione che prevede una fase di identificazione attraverso l'inserimento di alcuni dati anagrafici. Il cittadino o turista potrà navigare in tutte le reti federate distribuite sul territorio nazionale, e nei nuovi punti di accesso che saranno posizionati con particolare attenzione ai luoghi del turismo e della cultura. L'accesso alla rete avverrà in modalità automatica e trasparente al cittadino: cioè senza che questi si debba nuovamente autenticare. Questo approccio da un lato garantisce semplicità per il fruitore, che ha un unico sistema di accesso automatizzato, dall'altro rappresenta un vantaggio anche per i gestori delle reti federate poiché possono dare accesso ad altri utenti senza dover richiedere nuove registrazioni. Grazie alla App sarà inoltre possibile usufruire di servizi centralizzati o locali quali informazioni, servizi di acquisito biglietti e/o consultazione di contenuti, ecc. Al momento la APP è realizzata nella versione IOS e Android, si sta valutando anche la possibilità di una prossima realizzazione nel sistema operativo WindowsPhone.
Sebbene vi sia un deciso impegno da parte dello stato , recenti test hanno mostrato come tale servizio sia ancora acerbo: Wi-Fi-Italia nasce come programma di copertura internet per quelle zone del Paese in cui gli operatori non arrivano o non investono per mancanza di mercato. E tecnicamente questo è il suo vantaggio, economico e funzionale (se crolla un nodo non crolla la rete), e anche il suo limite: non si tratta di grandi network wireless per aree pubbliche, ma una "federazione" di piccoli router messi "a disposizione", uniti da una rete autenticata dall'app.Al momento , non solo questi router sono presenti in numero molto ridotto, specie nelle zone del sud Italia, ma , come è stato verificato , funzionano soltanto all'interno delle infrastrutture che ospitano il nodo di connessione alla rete.

## 1.2 Il tema della legalità

Il tema della condivisione di reti WiFi al pubblico ha sollevato non poche problematiche dal punto di vista legislativo.
Sebbene l'obbligo di identificazione per coloro i quali utilizzavano una rete pubblica fosse già caduto nel 2011, con la scadenza di alcuni termini del decreto Pisanu, serviva una norma che per prima cosa esplicitasse questo principio e che poi facesse piazza pulita anche di altri obblighi per gli esercenti che offrivano il Wi-Fi: sia quelli del codice delle comunicazioni (che valgono per i provider di internet) sia quelli sopravvissuti nel Pisanu contro il terrorismo. Dal 2013 quindi un esercente, un negozio, un hotel, un ristorante, ma anche una pubblica amministrazione può liberalmente installare un hot spot, collegarlo alla rete e offrire il servizio senza dover tracciare gli utenti, le loro connessioni, fornire account e password, né chiedere autorizzazioni. Il precedente testo del Fare invece chiedeva di tracciare i codici del dispositivo usato per la connessione (computer, tablet o cellulare) imponendo oneri tecnici e burocratici gravosi per qualsiasi esercente.[6]

*"L'offerta di accesso alla rete internet al pubblico tramite rete WIFI non richiede l'identificazione personale degli utilizzatori. Quando l'offerta di accesso non costituisce l'attività commerciale prevalente del gestore del servizio, non trovano applicazione l'articolo 25 del codice delle comunicazioni elettroniche di cui al decreto legislativo 1 gennaio 2003, n.259 e successive modificazioni, e l'articolo 7 del decreto-legge 27 luglio 2005, n. 144, convertito, con modificazioni, dalla legge 31 luglio 2005, n. 155, e successive modificazioni".*

Tuttavia, è bene ricordare che chi mette a disposizione di terzi la connessione a Internet intestata a sé o alla propria azienda può risultare responsabile, di attività illecite commesse tramite la connessione stessa, a meno che non sia in grado di dimostrare che siano stati terzi (anche sconosciuti) a commettere tali attività.
Tenere traccia di chi si connette alla rete risulta quindi buona norma : tramite operatore, tramite l'access point o tramite autenticazione risalire all'utilizzatore potrebbe sempre tornare utile.[7] In primo piano pertanto

spicca la questione della tutela e della sicurezza per coloro i quali decidano di condividere la propria rete: può servire per discriminarsi, nei confronti di indagini di polizia, qualora qualche utente utilizzi la connessione offerta per commettere reati. In altri Paesi europei è capitato che l'esercente fosse considerato corresponsabile, in questo caso. In Germania come citato dal Sole 24 ore , una sentenza del maggio 2010 ha dichiarato parzialmente responsabile il proprietario/utente di una rete Wi-Fi che non abbia utilizzato adeguati sistemi di protezione dal rischio di utilizzi abusivi della connessione per finalità illecite. Il caso riguardava lo scambio di file pirata. Stessa casistica nel Regno Unito, dove però è proprio il Digital Economy Act a imporre che siano identificati gli utenti che violano il copyright. La Francia addirittura chiede di tenere per 12 mesi il registro delle connessioni e di fare il possibile per consentire di risalire all'identità degli utenti.In Italia la normativa non è così esplicita e non c'è una giurisprudenza chiara, in merito. Però già adesso, e da tempo, le principali reti Wi-Fi identificano in modo sicuro gli utenti, via sim del cellulare, quindi il problema è marginale.

Specie in Italia, un ruolo fondamentale è giocato dagli ISP *Internet Service Provider*. In caso di illecito, nella misura in cui il proprietario della connessione non riesca a discolparsi identificando l'utilizzatore della connessione, accusato di aver commesso la violazione della legge, può in determinate circostanze, chiedere che vengano consultati i tabulati degli ISP al fine di identificare il colpevole.
Attualmente un ISP è obbligato alla conservazione dei dati relativi al traffico telematico per un anno. Tali dati comprendono data di inizio e fine connessione, indirizzo IP ma non il tabulato dei siti web visitati, come fino al 2008 veniva fatto da diversi providers.
Se già nel 2015 l'obbligo di conservazione dei dati era salito a 2 anni per poi tornare ad uno, ad oggi, per il crescente pericolo legato al terrorismo, la camera ha approvato un emendamento che porta l'obbligo da 12 a ben 72 mesi. Sarà necessario dunque stabilire un giusto contrappeso tra l'utilizzo dei dati e la gravità del reato al fine di non commettere abusi nella consultazione dei tabulati[8].
Tale provvedimento risulta quindi positivo dal punto di vista di colui che

offre la connessione, tuttavia, l'utilizzatore viene di fatto monitorato per 72 mesi, in una situazione che potrebbe, senza gli adeguati controlli, sconfinare nella violazione della privacy.

CAPITOLO 2

# *Progettazione*

# 2 Progettazione

## 2.1 Descrizione del progetto di tesi

Il progetto di tesi Apostrofe consiste nel realizzare una applicazione per dispositivi mobili Android che consenta all'utente di registrare il proprio access point presso l'applicazione condividendo così la propria connessione con gli altri utenti che utilizzano APostrophe . Obbiettivo della tesi è anche l'implementazione di un sistema di identificazione che verifichi l'effettiva appartenenza del router ad un determinato utente preservando allo stesso tempo la password di connessione al dispositivo e un insieme di servizi per usufruire della connessione.

## 2.2 Architettura del sistema

APostrophe si presenta come un'applicazione client-server in cui i dati vengono memorizzati su un database presente in rete e su un database locale all'interno di ogni dispositivo su cui l'applicativo è installato . L'architettura del sistema prevede una struttura client abbastanza complessa dove sono state implementate la maggior parte delle funzionalità, una struttura ben più semplice per quel che concerne il lato server e il database.
Tramite l'applicazione installata sul dispositivo (**lato client**), l'utente invia richieste al **server**, relativamente ai dispositivi mappati o ai propri dati personali , il server interrogando il **database** restituisce le informazioni ottenute al client che le mette a video.

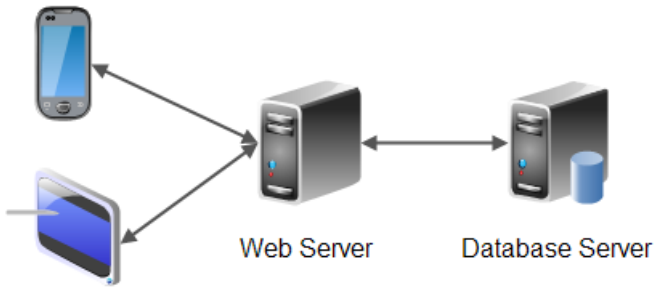


Figura 1: Struttura del progetto

## 2.3 Tecnologie utilizzate

In questa sezione è descritta a livello macroscopico la struttura di APostrophe e il linguaggio impiegato per definirla.

- **Client** : progettato per sistemi Android fa uso di due linguaggi fondamentali: Java e XML. Tramite **Java** è stato realizzato lo scheletro dell'applicativo, l'algoritmo di posizionamento, quello di connessione ai dispositivi , il collegamento tra UI e sistema. Sono scritte in Java anche le classi necessarie per la chiamata HTTP al web server. XML invece è stato impiegato per la realizzazione e personalizzazione di tutta l'interfaccia utente.

- **Server**: il server ( che si appoggia ad Altervista) è stato interamente scritto in **PHP** , è funzionale a stabilire una connessione con il database e ad eseguire le eventuali query. Le risposte da server a client possono essere stringhe semplici oppure, in caso di strutture complesse, vengono rappresentate con il formato **Json**.

- **Database**:viene suddiviso in database locale e database in rete. Il database locale è scritto in **SQLite** , una versione semplificata di MySQL e si interfaccia con il client attraverso una libreria Java . Il

database in rete è scritto in **MySQL** e viene interrogato dal server tramite libreria PHP.

Strumenti utili allo sviluppo sono stati **Android Studio** (ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android) , **textEditor Altervista** (scrittura classi PHP) , **phpMyAdmin** , un'applicazione web scritta in PHP che consente di amministrare un database MySQL o MariaDB tramite un qualsiasi browser.

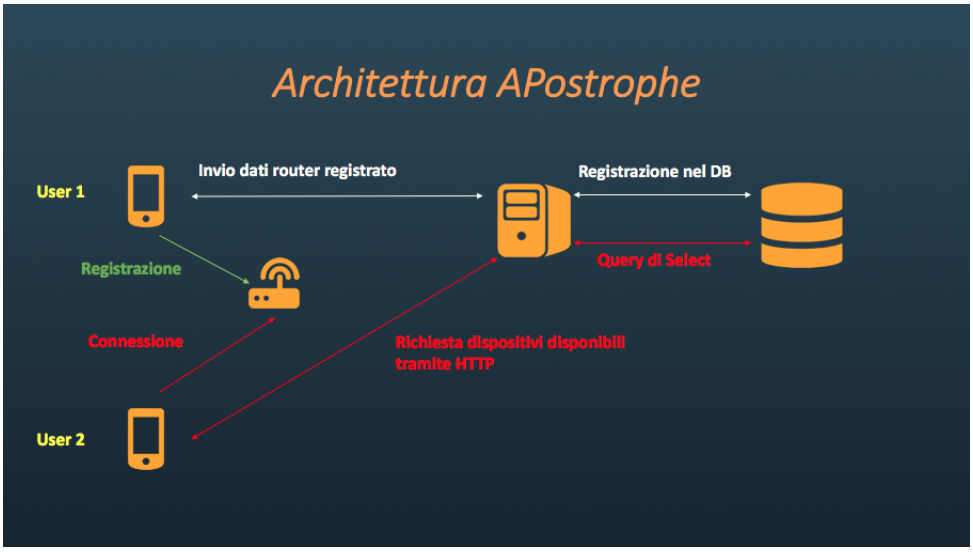


Figura 2: Architettura APostrophe

Le tecnologie utilizzate sono state scelte secondo un precisi criteri, tuttavia, il problema principale che è alla base dello sviluppo di un applicativo resta sempre : nativo o ibrido? Come si può constatare, sebbene si siano presi in considerazione, framework alternativi, l'applicativo (escludendo lato server e UI) è interamente progettato in Java. La scelta deriva da una maggiore conoscenza del linguaggio e da una elevata presenza di materiale

in rete che offriva snippet di codice o addirittura intere classi java facilmente riadattabili alle esigenze dell'applicativo come ad esempio l'implementazione del sistema di cifratura per la trasmissione di dati su HTTP ma anche la stessa classe per le chiamate. Anche la presenza dell'API WiFi Manager e la sua completezza ha influito molto sulla scelta del linguaggio.
Per i database, i linguaggi sono stati invece vincolati dalla scelta della collocazione della base di dati stessa e dalla disponibilità delle librerie. SQLite si configura perfettamente con Java ed è la miglior soluzione come database locale su piattaforma Android, l'utilizzo di un database MySQL è stato invece imposto dalla scelta di optare per Altervista come gestore per l'implementazione del server. I servizi di Altervista comprendono infatti oltre ad uno spazio in cui creare il server , anche di tool integrati per la gestione di un database relazionale MySQL.
Alla luce del percorso didattico svolto nei tre anni rappresentano comunque la tipologia di database più utilizzata e di cui si dispongono maggiori conoscenze. La struttura del database è molto semplice , conta appena 2 tabelle strutturate come segue[9] :

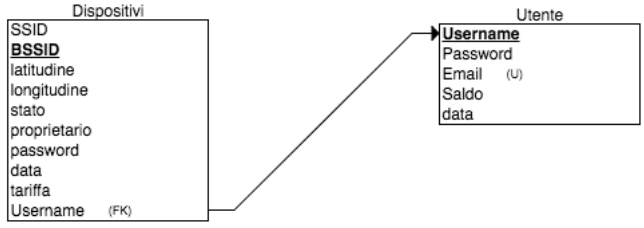


Figura 3: Tabelle database

I campi di ogni tabella assolvono le seguenti funzioni:

Tabella **Dispositivi**:

- **SSID**: "nome" della rete WiFi , modificabile dall'utente.
- **BSSID**: MAC address del dispositivo , univoco e mai modificabile.
- **latitudine**: coordinata ottenuta in fase di posizionamento.
- **longitudine**: coordinata ottenuta in fase di posizionamento.
- **stato**: può essere *disponibile* o *cancellato* e fa riferimento allo stato del router che, anche in caso di eliminazione, rimane sul database, in caso di nuova registrazione viene sovrascritto.
- **proprietario**: colui che ha registrato il router.
- **password**: password per ricevere la connessione , modificabile dall'utente
- **data**: ultima data di modifica dell'elemento in tabella , necessario per i download delle informazioni sugli altri smartphone (spiegato dettagliatamente nel capitolo 3 ).
- **tariffa**: cent/min che l'utente cliente dovrà pagare per usufruire della connessione di quel particolare dispositivo.

Tabella **Utente**:

- **Username**: username di accesso dell'utente , univoco e modificabile dall'utente.
- **Password**: password di accesso dell'utente , modificabile modificabile dall'utente.
- **Email**: email di registrazione necessaria ad identificare l'utente, univoca e modificabile dall'utente.
- **Saldo**: credito caricato dall'utente.

- **data**: data di modifica del particolare utente.

È importante evidenziare che la struttura tabellare non è la stessa su entrambi i database (locale e in rete). Infatti, il database locale non ha motivo di gestire la tabella utenti, la quale sarebbe superflua, poiché è necessario consultare il database online per sapere se l'username e l'email utilizzate, qualora ci si voglia registrare, siano corrette, oppure bisogna verificare se username e password utilizzate per il login siano validi. Gestire questo database anche in locale creerebbe problemi di concorrenza affatto trascurabili (si immagini una modifica del nome utente qual ora l'utente sia loggato su più dispositivi e non tutti connessi alla rete), pertanto si è preferito optare per una gestione locale tramite **Shared Preferences** , spiegato in dettaglio nel capitolo 3.

### Cifratura dei dati : AES

Dovendo manipolare informazioni personali degli utenti e dovendo gestire password di accesso, non si può tralasciare l'aspetto della sicurezza. I dati, in qualsiasi momento, potrebbero essere carpiti da un potenziale malintenzionato, che avrebbe così accesso non solo alle informazioni personali degli utenti, ma a tutte le password degli hot spot registrati. Per prevenire tale evento, sono stati individuati tre possibili momenti in cui il sistema andrebbe messo in sicurezza :

1. quando i dati risiedono sullo smartphone;
2. quando i dati sono nel database in rete;
3. nel passaggio tra client e server.

Nel primo caso, un potenziale attacco ad un sistema sprovvisto di sistemi di sicurezza può verificarsi semplicemente possedendo i permessi di root sul dispositivo sul quale è installata l'applicazione. In questa situazione, si potrebbero visualizzare in chiaro tutti i dati su database SQLite, ed avere facilmente accesso alle password di tutti i router .
Il secondo caso, più remoto, ma sempre possibile, è il caso in cui vengano

violate le protezioni sul server : i dati sul database sono sempre in chiaro . Il terzo caso, infine, è il più probabile , e mira a carpire informazioni private con il classico sistema del "man in the middle".
Per ovviare al primo e secondo caso , al fine di garantire una soglia minima di sicurezza, sarebbe sufficiente criptare i dati prima che vengano inseriti nel database con un sistema di cifratura ritenuto sicuro.
Essendo APostrophe ancora in via di sviluppo , si è scelto di cifrare, momentaneamente, solo i dati che viaggiano in chiaro durante il terzo caso analizzato: per tale scopo ci si è avvalsi dell'algoritmo **AES** (Advanced Encryption Standard).

AES , conosciuto anche come Rijndael [10][11], è un algoritmo di cifratura a blocchi utilizzato come standard dal governo degli Stati Uniti d'America. L'algoritmo è stato sviluppato da due crittografi belgi, Joan Daemen e Vincent Rijmen, che lo hanno presentato al processo di selezione per l'AES con il nome di "Rijndael", nome derivato dai nomi degli inventori.
Sebbene si è soliti identificare AES anche come "Rijnadael" formalmente risultano leggermente differenti , o meglio, possiamo definire AES come "Rijndael" a 128 bit.
AES opera utilizzando matrici di 4 X 4 byte chiamate **stati** . Per cifrare, ogni round (ciclo) dell'AES (eccetto l'ultimo) consiste nei seguenti quattro passaggi:

1. *SubBytes*: ogni byte della matrice viene modificato tramite la S-box (o Substitution box) a 8 bit. Questa operazione provvede a fornire la non linearità all'algoritmo.

2. *ShiftRows*: provvede a scostare le righe della matrice di un parametro dipendente dal numero di riga.La prima riga è invariata, la seconda viene spostata di un passo verso sinistra, la terza di due posti e la quarta di tre posti.

3. *MixColumns*:vengono presi quattro byte di ogni colonna e combinati utilizzando una trasformazione lineare invertibile. Ogni colonna è trattata come un polinomio e moltiplicata per un polinomio fisso.

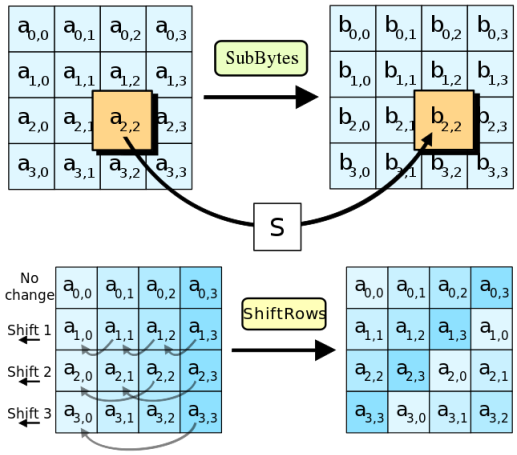


Figura 4: SubBytes e ShiftRows

4. *AddRoundKey*: vengono combinate con un XOR la chiave di sessione con la matrice ottenuta dai passaggi precedenti. La chiave di sessione viene ricavata dalla chiave primaria dal gestore della chiave.

Questi passaggi sono ripetuti 10 volte ottenendo così il cifrario di crittazione. Quello di decrittazione è ugualmente ottenibile scambiando però l'ordine dei passaggi sopra elencati. Dal 2001 sono stati pubblicati diversi progetti di attacco ad AES basati su metodi esaustivi o nel migliore dei casi su algoritmi 4 volte più veloci degli esaustivi. I tempi richiesti per recuperare la chiave sono comunque elevatissimi; nel 2011 è stato pubblicato Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger (2011). Biclique Cryptanalysis of the Full AES che richiede qualcosa come 2126 operazioni per forzare una chiave AES a 128 bit, 2190 per AES 192 bit e 2254 per AES a 256 bit. Il livello di sicurezza di AES sembra quindi molto elevato; per i documenti del governo USA AES 128 bit è considerato sufficiente per i documenti classificati Secret, mentre per i Top secret occorre AES 192 o meglio ancora 256 bit.

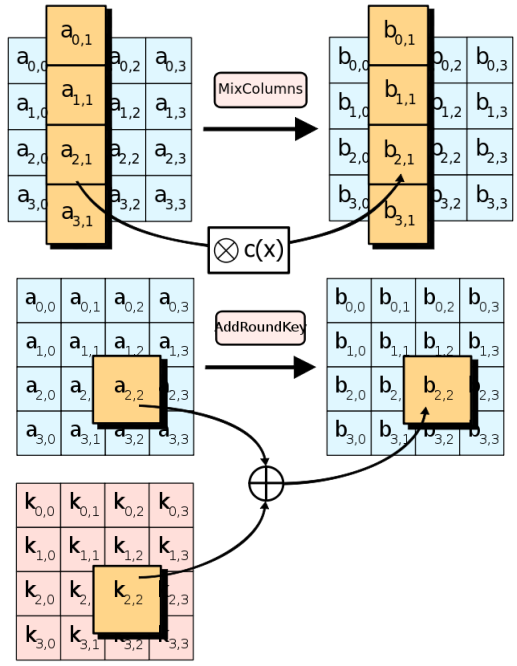


Figura 5: MixColumns e AddRoundKey

## 2.4 Funzionalità

Sebbene il progetto APostrophe sia ancora in via di sviluppo , l'obbiettivo di questa tesi resta comunque quello di visualizzare una prima versione del prodotto funzionante. Di seguito vengono descritte le funzionalità di cui dispone la versione "alfa" dell'applicativo.

### 2.4.1 Menù e componenti fondamentali

Una volta selezionata l'applicazione nel menù Android l'utente verrà indirizzato sulla pagina principale dalla quale potrà scegliere se effettuare il login o registrarsi. Qualora avesse già effettuato il login precedentemente, verrà indirizzato direttamente nella propria home.

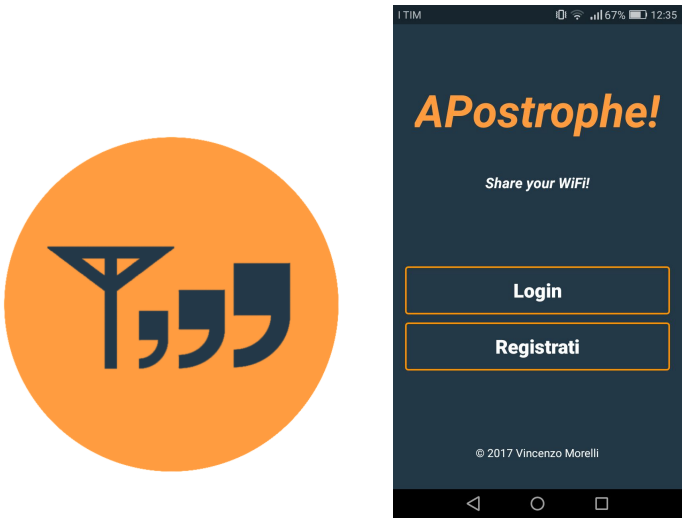


Figura 6: Logo e schermata di Login

La **Home** permetterà di accedere alle mappa dei dispositivi attraverso il pulsante "Scan" , in alternativa sarà possibile esplorare il menu. Quest'ultimo, oltre che dall'opzione "Home" è composto dalle seguenti voci:

- Profilo: consente di visualizzare ed eventualmente modificare i dati personali dell'utente , quali username, password, Impostazioni:consenteemail;
- Gestisci Dispositivi: permette di gestire i diversi dispositivi registrati, modificandone eventualmente SSID o password di accesso alla connessione, o di registrarne di nuovi o eliminare vecchi dispositivi;
- Impostazioni: consente di visualizzare il saldo del conto personale dell'utente ed eventualmente ricaricare di 5 o 10 euro tramite carta PayPal per poter effettuare pagamenti nel noleggio di una connessione.
- Logout : permette di effettuare il log out dal profilo utente.

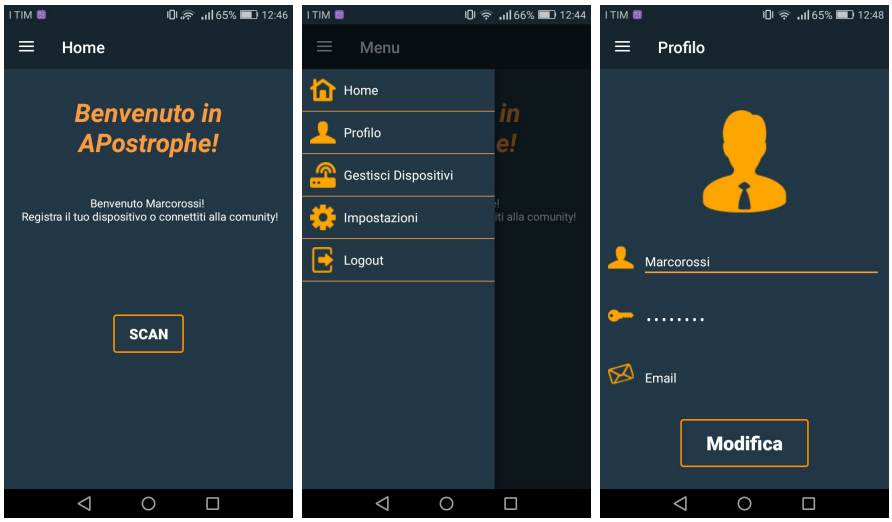


Figura 7: Schermate di home, menù e profilo utente

### 2.4.2 Mappatura del router

Una delle componenti principali di APostrophe è costituita dalla mappatura del router. Le caratteristiche su cui deve basarsi lo sviluppo dello scheletro di tale sezione sono **Riservatezza Integrità e Disponibilità**, per quel che concerne la salvaguardia dei dati forniti dall'utente. Affinché la community APostrophe cresca in fretta, è fondamentale che l'utilizzatore dell'applicazione sia invogliato a registrare il proprio dispositivo senza paura che qualcuno riesca a scoprirne la password con sistemi illeciti. Infatti, tramite il WiFi manager , disponendo dei permessi di root, è possibile visualizzare in chiaro la password del dispositivo a cui si è connessi. APostrophe, pertanto, restringe l'utilizzo dei servizi di navigazione solo all'interno dell'applicazione così, quando quest'ultima viene messa in pausa o arrestata si ha l'immediata cancellazione della password dal WiFi manager.
Tale restrizione è da considerarsi provvisoria: l'obbiettivo finale è quello di riuscire ad offrire una completa navigazione all'utente che sfrutta la con-


Figura 8: Schermate di gestione dispositivi e impostazioni

nessione.

Altresì importante è verificare che il router registrato che , possibilmente , costituirà, seppur marginalmente , una fonte di guadagno , appartenga alla persona che decide di registrarlo.
A tal proposito , per registrare l'hot spot, è richiesto il controllo fisico sul dispositivo: l'applicativo , infatti, genererà una password casuale per ogni dispositivo che si intende registrare. L'utente dovrà, quindi, accedere al pannello di controllo del router e modificare la password (con la possibilità di ripristinarla in seguito), in questo modo non basterà avere accesso alla rete per considerare proprio l'hot spot , sarà anche indispensabile possedere la password di accesso al pannello di controllo del dispositivo. Una volta effettuata la modifica, si potrà scegliere se lasciare la connessione completamente "free" oppure impostare un prezzo di utilizzo ( espresso in cent/min) . Premendo sul bottone "ok" si proverà l'accoppiamento con il dispositivo e, in caso di successo, si procederà al posizionamento. I dispositivi saranno

registrabili da un solo utente per volta. L'eliminazione dal database darà quindi la possibilità di "intestarlo" ad un altro membro della community.

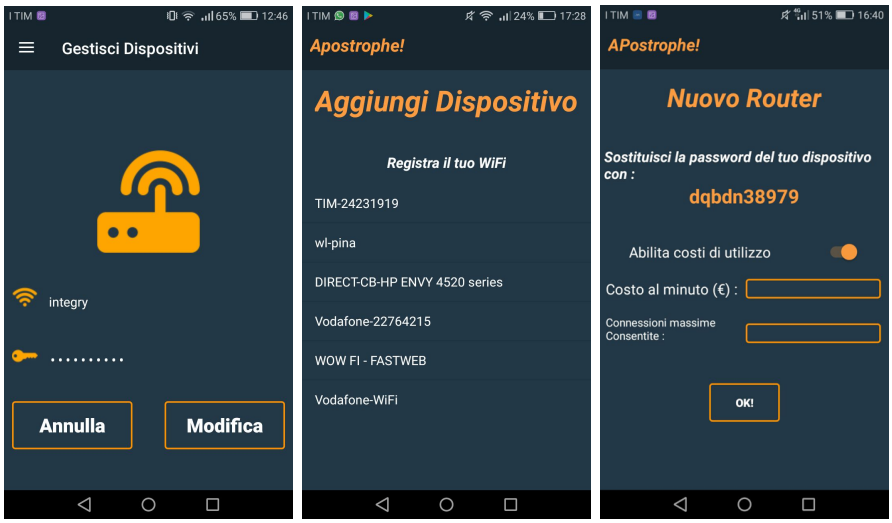


Figura 9: Schermate di modifica router, scelta del dispositivo, registrazione dispositivo

### 2.4.3 Servizi dell'applicativo

Aprendo la schermata di gestione dei dispositivi sarà possibile gestire i propri dispositivi tramite operazioni di modifica (SSID , password) oppure eliminazione .
Dopo aver mappato il dispositivo, esso, oltre ad essere disponibile nella sezione "Gestisci Dispositivi", verrà visualizzato sulla mappa e sarà a disposizione degli utenti che vorranno connettersi .
Selezionando , quindi , l'hot spot preferito sulla mappa si potrà effettuare la connessione specificando il tempo per il quale se ne ha bisogno ( in caso di hot spot a pagamento verrà accertato che il credito sia sufficiente).
Per questa prima versione si potrà scegliere se navigare 1min, 2min, 5min,

10min o 30min.
Adesso non resterà che esplorare l'area "Servizi" , selezionare il canale a cui si vuole accedere e navigare . Appena esaurito il tempo a disposizione la sessione termina e si ritorna nella pagina principale.

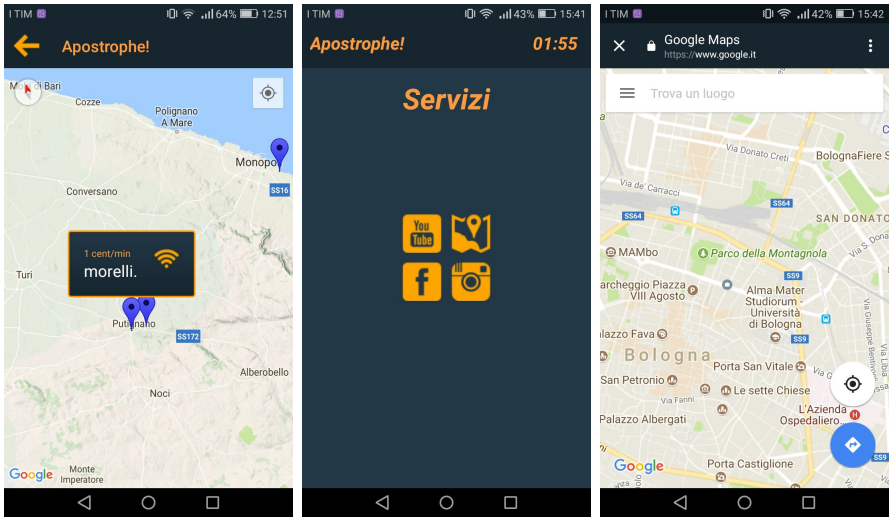


Figura 10: Schermate di selezione dispositivo, servizi e navigazione

Sarà sempre possibile ricercare connessioni e connettersi anche effettuando pagamenti , tuttavia, gli altri servizi , dal download dei nuovi dispositivi alla modifica di quelli registrati saranno utilizzabili esclusivamente in presenza di connettività alla rete.

CAPITOLO 3

# *Implementazione*

# 3 Implementazione

In questo capitolo verrà descritto in maniera dettagliata ogni algoritmo o strutta dati che è alla base degli aspetti caratterizzanti dell'applicativo.

## 3.1 Login e registrazione

Per poter usufruire dei servizi offerti da APostrophe è richiesta la registrazione all'applicazione. Gli scopi fondamentali per i quali è necessaria sono principalmente due.
Il primo, che può essere considerato quello più importante, è il bisogno di autenticazione.
Sebbene , come appurato nel capitolo 1 , dal 2011 fosse caduto l'obbligo di identificare coloro i quali hanno accesso a reti pubbliche (e dal 2013 anche quelli che hanno accesso a reti private), è sempre buona norma, per mettersi al sicuro da problemi giudiziari, tenere traccia di chi si connette alla propria rete, inoltre, un maggior controllo sugli accessi , accresce la fiducia degli utenti nel servizio , invogliando a mettere a disposizione il proprio access point, a partecipare attivamente alla vita della community.
In secondo luogo, la scelta di implementare un sistema di pagamento per l'utilizzo della connessione ha reso indispensabile la creazione di un account identificativo dell'utente.

La registrazione, è quindi possibile inserendo una password assieme ad un username e una mail non ancora presenti nel database, in caso di email già utilizzata o username già esistente l'account non viene creato. Per rendere più "user-friendly" l'applicazione, si è scelto di implementare un sistema di memorizzazione degli accessi, in questo modo, servirà effettuare il login solo la prima volta, successivamente si verrà indirizzati direttamente sulla home. Android per realizzare questa *feature* mette a disposizione il sistema delle **Shared Preferences**.[12]

### Il sistema delle Shared Preferences

In presenza di valori chiave da salvare e consultare rapidamente, le SharedPreferences APIs costituiscono la scelta più oculata. Un oggetto SharedPreferences punta ad un file in cui sono salvate le coppie chiave-valore delle informazioni registrate. Lettura e scrittura delle SharedPreferences sono semplici e immediate, totalmente gestite dalle API SharedPreferences. Vi è inoltre la possibilità di renderle pubbliche o private a seconda dell'utilizzo, la differenza è che una coppia chiave valore *shared* , ossia pubblica , è utilizzabile non solo dall'APP che la ha creata ma da tutte le APP nel dispositivo.
Si dimostra un sistema di salvataggio efficiente quando vi sono pochi dati da memorizzare e da consultare rapidamente. Per gestire quantità di dati considerevoli è necessario l'utilizzo di un database vero e proprio.

Listing 1: Memorizzazione dell'username

```
editor= getSharedPreferences("Utente", MODE_PRIVATE).edit();
editor.putString("Username",username );
editor.commit();
```

Dopo aver memorizzato l'username, ad ogni avvio dell'applicazione viene verificato che nella file di Sharedpreferences sia contenuto un username, in caso positivo avviene il reindirizzamento alla homepage.

Listing 2: Verifica e reindirizzamento

```
prefs=getSharedPreferences("Utente", MODE_PRIVATE);
username = prefs.getString("Username", null);
if(username!=null){
        Intent m=new Intent(this,MenuActivity.class);
        startActivity(m);
        finish();

    }
```

## 3.2 Gestione e mappatura dispositivi

Una volta arrivati nella home, attraverso il menù di navigazione, è possibile accedere alla sezione di gestione dispositivi.
Per l'inserimento del primo hotspot personale è sufficiente posizionarsi in prossimità dell'access point(abbastanza vicino da poter essere rilevato) e cliccare sul bottone "+" , così facendo, viene aperta una interfaccia nella quale sono presenti tutte le connessioni WiFi circostanti e non registrate all'interno dei database. Selezionando la connessione desiderata si procede alla configurazione.

### 3.2.1 Appartenenza del dispositivo

APostrophe si propone come una community in cui gli utenti possano condividere il proprio dispositivo ed utilizzare quelli condivisi dagli altri, anche tramite il pagamento di una somma pari al costo per minuto deciso dal proprietario dell'hot spot.
Frequentemente capita di possedere password di reti WiFi di cui non si è proprietari. Se non viene posta sotto controllo questa situazione, ogni utente potrebbe condividere **come suo**, un qualunque access point soltanto conoscendone la password, senza chiedere consensi al proprietario e per giunta guadagnando da questa attività.
Per ovviare al problema, è stato implementato un metodo che consente di accertare che l'utente sia il vero proprietario del dispositivo.

Per registrare il router, infatti, è richiesto all'utente di modificare la password della connessione WiFi stessa, così da certificare un controllo maggiore sul dispositivo, poiché, per effettuare tale operazione è necessario conoscere le credenziali d'accesso del pannello di controllo dell'access point che, di norma, sono esclusivamente in possesso del proprietario.
La password da inserire è dettata dal sistema , il quale effettua anche distinzione tra password, **WEP** e **WPA2 o WPA-PSK**. Il WiFi Manager ricopre un ruolo di primo piano per il riconoscimento della connessione. In questo caso tramite l'analisi della rete è possibile comprenderne le capabilities ed identificare il diverso tipo di connessione con conseguente generazione di una nuova password adatta al sistema di sicurezza selezionato.

### 3.2.2 WiFi Manager : presentazione

Android consente alle applicazioni di avere accesso allo stato della connettività WiFi a livelli molto bassi, pertanto, è possibile accedere, utilizzare ed eventualmente modificare la maggior parte delle informazioni relative ad una connessione.
Le informazioni utilizzabili possono includere SSID, MAC address, sistema di protezione, ma anche frequenza del segnale ed intensità.
L'API WiFi Manager [13]consente, inoltre, di stabilire una nuova connessione , interromperla, aggiungere configurazioni di rete o rimuoverle tra quelle già salvate.

### 3.2.3 WiFi Manager: individuazione delle reti

Per l'individuazione delle reti wifi , l'activity che deve visualizzare la lista delle connessioni chiama il servizio *WiFiConnect* . La chiamata di un servizio che effettua la scansione delle reti disponibili implica la presenza di un *broadcast receiver* registrato presso l'activity chiamante.
Il *broadcast* non è altro che un evento provocato o scaturito dal sistema. Quando un *broadcast* è inviato, esso viene automaticamente veicolato dal sistema verso le app che hanno dichiarato di voler ricevere quel particolare tipo di broadcast In generale, possono essere utilizzati come sistema di

scambio di informazioni interne al sistema e lontane dal normale flusso di dati dell'utente. Tuttavia, è bene non abusare dei *broadcast* perchè potrebbero influire negativamente sulle performance del sistema.
Il *receiver*, registrato per utilizzare la lista di connessioni ottiene i dati inviati dal servizio e li elabora in maniera **asincrona**. Naturalmente, di ogni connessione rilevata, è possibile selezionare ed inviare solo i dati che servono nella particolare circostanza .

Listing 3: Ricerca dispositivi

```
@Override
   public void onCreate() {
       super.onCreate();
       mainWifi = (WifiManager)
           getSystemService(Context.WIFI_SERVICE); //dichiaro il
           WiFi Manager
       mainWifi.startScan(); //effettuo una scansione
       wifiList = mainWifi.getScanResults(); //prendo i risultati

       for (ScanResult res : wifiList) { // costruisco l'oggetto
           WiFi prendendo solo i campi interessati

           String stato =new String();
           double distanza =(calculateDistance((double) res.level,
               res.frequency));

           WiFi  oggWiFi=new
               WiFi(res.SSID.toString(),res.BSSID.toString(),distanza,stato,0);

           wifi.add(oggWiFi);
       }

       Intent i = new Intent("wifireciever");

       i.putParcelableArrayListExtra("wifi",wifi);
       sendBroadcast(i); // invio la risposta

   }
```

```
}
```

Di norma, l' operazione di *scan* richiederebbe di per se l'implementazione di un *broadcast receiver* che, ad ogni scansione riceve una nuova lista di risultati. Poichè l'operazione di *scan* è inserita in un servizio rilanciato volta per volta, è sufficiente effettuare uno solo *scan*, raccogliere i dati in una *WiFi list* e, dopo averli manipolati, rispedirli all'attività chiamante sotto forma di *arraylist* di connessioni, normali oggetti Java (Listing 3).

### 3.2.4 WiFi Manager: connessione e gestione eccezioni

Dopo aver selezionato il dispositivo e aver cambiato la password della connessione con quella indicata da APostrophe si procede alla registrazione del dispositivo.
Quest'ultima si divide in 2 fasi:

- Connessione alla rete;
- Posizionamento del dispositivo sulla mappa (spiegato nella sottosezione successiva).

Per stabilire una connessione con un particolare hot spot , non è sufficiente sfruttare le funzionalità dell'API WiFi Manager, ma è richiesta una gestione asincrona degli **stati di connessione** che non sempre seguono uno schema regolare. Questa volta, è necessaria l'implementazione di un *broadcast receiver* presso il quale il WiFi Manager registra "l'intento" di cambiare stato al fine di poterlo gestire (Listing 4).

Listing 4: Registrazione dell'intent presso il broadcast

```
//creo un nuovo broadcast receiver responsabile di "catturare"
    intent lanciati dal sistema

     BroadcastReceiver broadcastReceiver = new
         WifiBroadcastReceiver();

    //definisco gli intent da lanciare come cambi di stato del
        WiFi manager
     // e li registro presso il broadcast receiver

     intentFilter.addAction(WifiManager.SUPPLICANT_STATE_CHANGED_ACTION);

     this.registerReceiver(broadcastReceiver, intentFilter);
   }
```

In un' *happy path* ideale , inizialmente, il WiFi non è collegato ad alcuna rete ( *Disconnected* ), si procede, pertanto, con la scansione (*Scanning*) dei dispositivi fino a quando non viene rilevato quello per cui è stata richiesta la connessione, viene tentata l'associazione (*Associating*), e, completato quest'ultimo passaggio (*Associated*), si passa ai controlli di sicurezza *Four-Way-Handshake, Group-Handshake*, prima di dichiarare il processo concluso con successo (*Completed*).
Solitamente, la situazione precedentemente descritta accade molto raramente e, le responsabilità possono essere attribuite talvolta all'utente, ma spesso anche all'API di Android.
L'utente, per esempio, potrebbe commettere errori di inserimento password oppure potrebbe uscire dall'area di copertura della rete WiFi durante il procedimento. Il WiFi manager, d'altra parte, se precedentemente connesso ad un'altra rete, non parte dallo stato *Disconnected*, ma da *Completed*, quindi, è necessario che quest'ultimo non venga considerato nella computazione; ancora, potrebbe capitare che non riesca ad associarsi al dispositivo immediatamente e ripeta il passaggio *Disconnected -Associating* un numero indefinito di volte, comportamento che si palesa anche in caso di inserimento di una password errata.

Listing 5: Verifica range e metodo di sicurezza

```
// verifico che sia presente tra i router a portata e se non lo è
    più genero un errore
for (ScanResult res : wifiList) {

     //verifica che la connessione sia in range
      if(connetti.equals(res.SSID)){

          isCovered=true;
          bssid=res.BSSID;

           // assegnazione delle chiavi in base al metodo di
               autenticazione
          // distinzione tra WPA e WEP tramite il parametro
              CAPABILITIES
          if(res.capabilities.toString().substring(0,4).equals("[WPA")){

              conf.preSharedKey = "\""+passwordWPA+"\"";
              pass.setText(passwordWPA);
              break;

          }else
              if(res.capabilities.toString().substring(0,4).equals("[WEP"))
              {

               conf.wepKeys[0] = "\"" + passwordWEP + "\"";
               conf.wepTxKeyIndex = 0;
               conf.allowedKeyManagement.set(WifiConfiguration.KeyMgmt.NONE);
               conf.allowedGroupCiphers.set(WifiConfiguration.GroupCipher.WEP40);

               check=1;
               pass.setText(passwordWEP);
               break;

          }else if(res.capabilities.toString().equals("[ESS]")){
```

```
                check=0;
                conf.preSharedKey = "\""+passwordWPA+"\"";
                pass.setText(passwordWPA);
                break;
            }
        }

}
        //se non sono in range genero un errore

    if(isCovered==false){

            AlertDialog.Builder builder;
            if (Build.VERSION.SDK_INT >=
                Build.VERSION_CODES.LOLLIPOP) {

                builder = new AlertDialog.Builder(new
                    ContextThemeWrapper(Inserimento.this,
                    R.style.AlertDialogCustom));

            } else {

                builder = new AlertDialog.Builder(Inserimento.this);

            }

            builder.setTitle("Errore!")
            .setMessage("Sei fuori dal range!")
               .setPositiveButton(android.R.string.yes, new
                   DialogInterface.OnClickListener() {

               public void onClick(DialogInterface dialog, int
                   which) {

                    finish();
               }

             })
```

```
            .show();

      }
}
```

Nel listato 5 è dimostrato come si è scelto di ovviare ai possibili errori di range , ovvero effettuando una scansione ulteriore e verificando che la connessione scelta sia presente nella WiFi list dei router scansionati (processo di scansione identico a quello in listato 3) tramite l'utilizzo di una variabile *boolean*.
Se questi errori non si presentano, è possibile effettuare la connessione sempre dopo aver modificato la password del dispositivo:

Listing 6: Connessione alla rete

```
public void connetti(){

        mainWifi.setWifiEnabled(true);
        mainWifi.addNetwork(conf);

        // prendo tutte le connessioni, le esamino , mi disconnetto
        // abilito la rete designata e provo a connettermi

        List<WifiConfiguration> list =
            mainWifi.getConfiguredNetworks();

        for( WifiConfiguration i : list ) {

            if(i.SSID != null && i.SSID.equals( conf.SSID )) {

                mainWifi.disconnect();
                mainWifi.enableNetwork(i.networkId, true);
                mainWifi.reconnect();
                break;
            }
         }
```

```
        //settando il controllo ad 1 confermo l'intento di voler
            stabilire
        // la connessione

           checkRete =1;
    }
```

Quando il bottone di connessione viene premuto , il *broadcast receiver* comincia a ricevere dal WiFi Manager gli stati relativi alla connessione.
Gli errori che possono presentarsi a questo punto sono gestibili tramite semplici controlli: se la password inserita è errata , è sufficiente contare i *Disconnected* per capire che all'incirca al quinto tentativo è necessario interrompere l'operazione, in questo caso , il dispositivo cercherà una rete già memorizzata a cui connettersi , anche non necessariamente quella designata . Lo stato mostrato sarà *Completed* ma di fatto il dispositivo risulterà connesso ad un altro WiFi, per cui l'implementazione di tale funzionalità richiede un ulteriore controllo.
Per concludere l'analisi su questa componente, è utile soffermarsi ancora sull'aspetto di sicurezza e sul metodo utilizzato per la rimozione di una rete. Si era detto che i servizi di APostrophe sono limitati all'interno dell'applicativo per evitare che chiunque possa visualizzare la password del router cui è connesso, in caso di chiusura o messa in pausa dell'applicativo, la rete viene istantaneamente rimossa. La questione è risultata piuttosto spinosa ed ha individuato una *"debolezza" dell'API*, dato che per dimenticare una rete già memorizzata prima dell'utilizzo dell'app è obbligatorio rimuoverla **manualmente** tramite il WiFi Manager dello smartphone, in caso contrario, la rete non viene eliminata del tutto, vengono cancellate le nuove credenziali inserite durante l'utilizzo dell'app però la vecchia configurazione persiste nella memoria del telefono anche se i comandi di rimozione sono stati eseguiti correttamente.

### 3.2.5 L'algoritmo di posizionamento

Dopo aver stabilito la connessione, il dispositivo deve essere posizionato sulla mappa per essere utilizzato della community.
A questo scopo, l'algoritmo di posizionamento fornisce, con buona approssimazione, le coordinate geografiche del router .
Esso si basa su un assunto fondamentale :

*L'intersezione di tre circonferenze non concentriche è data da uno ed un solo punto.*

Il centro delle circonferenze *Ci* è dato dalla posizione dell'utente al tempo *Ti* , il raggio *Ri* è la distanza dal router al tempo *Ti*.

Per la posizione dell'utente si utilizzano i servizi delle mappe di google (spiegati in seguito), la distanza dal router è calcolata elaborando i dati forniti dal WiFi manager ovvero la **frequenza** e il **livello** di intensità del segnale.

Listing 7: Calcolo della distanza in metri

```
double exp = (27.55 - (20 * Math.log10(freqInMHz)) +
    Math.abs(levelInDb)) / 20.0;

double val=Math.pow(10.0, exp);
```

Come si può osservare nel listato 7, il valore della variabile *val* non è altro che la distanza dal router in metri.
Sebbene sia corretta nel suo insieme, dai numerosi esperimenti effettuati è risultato che la formula non si può considerare del tutto attendibile, poichè non considera i disturbi (pareti e campi magnetici) che potrebbero indebolire il segnale.
È stato dimostrato che posizionandosi sempre più lontano dal dispositivo l'errore sul calcolo della distanza cresce in maniere **esponenziale**.
Per ovviare al problema, le distanze sono state suddivise in intervalli re-

golari e divise per una costante sempre maggiore che va a smussare il più possibile l'errore. Il metodo di calcolo non è comunque considerabile come certo.
Ad ogni posizione corrisponde una distanza dal router ( l'invocazione del metodo per il calcolo della distanza è osservabile nel listato 3) , i dati sono ora sufficienti per procedere.
Si considerino latitudine e longitudine rispettivamente come asse X e asse Y di un piano cartesiano. Per trovare il punto desiderato è sufficiente calcolare l'intersezione tra due delle tre circonferenze, i risultati possono essere due :

1. *Un punto di intersezione*: l'algoritmo ha concluso il suo lavoro poichè il valore trovato è unico ;

2. *Due punti di intersezione*: il router potrebbe trovarsi su uno dei due punti e bisogna analizzare la terza coppia di coordinate.

Nel secondo caso la situazione che si presenta è la seguente:

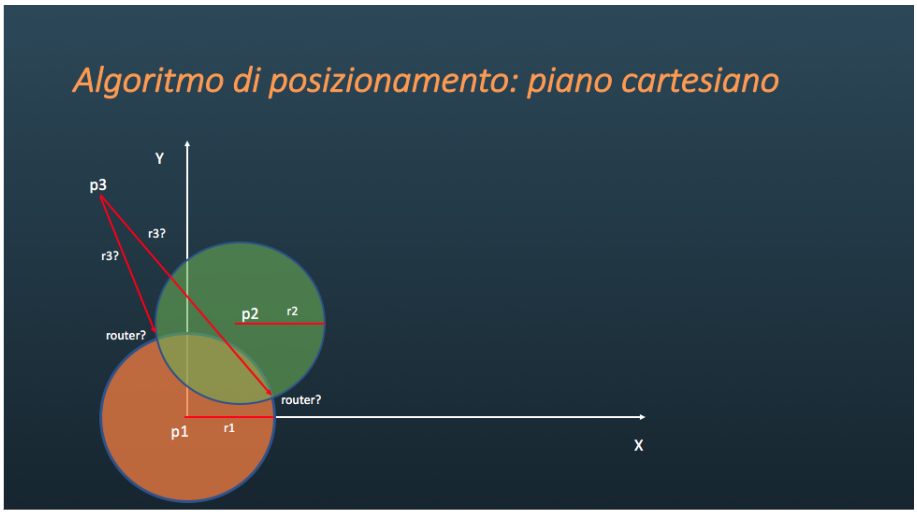


Figura 11: Piano cartesiano

Si noti che, per semplicità, le circonferenze sono state traslate verso l'origine per lavorare con valori più confidenziali. Il calcolo dell'intersezione fra le due circonferenze non è altro che la risoluzione di un sistema di equazioni di secondo grado, è bene sottolineare che, anche se le circonferenze, per come si presenta il problema, dovrebbero **sempre** incontrarsi in almeno un punto , questo spesso non succede perchè come già affermato in precedenza, l'algoritmo di calcolo delle distanze non è esatto e, in aggiunta, vi sono degli errori di posizionamento del GPS per quel che riguarda le coordinate. Essendo un posizionamento approssimato, si argina li problema risolvendo ricorsivamente il sistema di equazioni aumentando il raggio delle circonferenze fino a quando non si incontrano in almeno un punto. I dati sperimentali indicano che gli errori di posizionamento sono contenuti nel raggio di 10 metri.

Listing 8: Risoluzione del sistema e soluzioni coincidenti

```
// risolvo il sistema di equazioni ricavandomi delle costanti
       double E = (2 * pb - 2 * pd) / (-2 * pa + 2 * pc);
       double A = (pr * pr - pk * pk - pa * pa + pc * pc - pb * pb
           + pd * pd);
       double F = A / (-2 * pa + 2 * pc);

       // condizione necessaria ma , per il momento sempre
           verificata per non far azzerare i denominatori F ed E

       if (pa != pc) {

           a = E * E + 1;
           b = 2 * E * F - 2 * pc * E - 2 * pd;
           c = F * F + pc * pc - 2 * pc * F + pd * pd - pk * pk;

  // calcolo il delta

          d = b * b - 4 * a * c;

           if (d == 0) {

               s1 = (-b - sqrt(b * b - 4 * a * c)) / (2 * a);
               x1 = (s1 * E + F);

               //ho terminato il procedimento quindi ritraslo la
                   circonferenza avendo 1 come riferimento per il
               // primo e terzo caso e 2 come riferimento per il
                   secondo caso

               if(check==1 || check==0){

                   soluzione.add(x1*0.000001+val.get(0));
                   soluzione.add(s1*0.000001+val.get(1));}

               else if (check==2){
```

```
                soluzione.add(x1*0.000001+val.get(2));
                soluzione.add(s1*0.000001+val.get(3));

            }

            return soluzione;

        }
```

Risolvo il sistema di equazioni di secondo grado portando avanti nello sviluppo le costanti E , A ed F.
Vi è anche un controllo (per correttezza matematica) che considera l'opzione che il denominatore si annulli.
Si procede con il calcolo del delta e come dimostrato nel listato 8, qualora risultino soluzioni coincidenti, il punto di tangenza risulterà unico , pertanto, il dispositivo si troverà sicuramente in quel punto . Dopo aver traslato nuovamente le circonferenze ritorno quindi il risultato.

Listing 9: Intersezione fra circonferenza p1 e p2 e confronto con distanza da p3

```
 // se il delta è maggiore di 0 allora abbiamo 2 soluzioni reali
     distinte

if (d > 0) {

   s1 = (-b - sqrt(b * b - 4 * a * c)) / (2 * a);
   s2 = (-b + sqrt(b * b - 4 * a * c)) / (2 * a);

   x1 = (s1 * E + F);
   x2 = (s2 * E + F);

//posso pertanto calcolare il risultato per determinarlo considero
    i due punti di intersezione trovati , il centro della terza
    circonferenza non esaminata e la distanza di tale circonferenza
    dal router qualora la triangolazione fosse perfetta avrei che
    la distanza del mio punto dal router sarà uguale a quella da
```

```
    uno dei due punti in questo caso faccio la differenza tra le
    distanze e scelgo quella con il valore assoluto più vicino allo
    0

   if (check == 0) {

      if (sqrt(abs(Math.sqrt(Math.pow(dati.get(6) - x1, 2) +
          Math.pow(dati.get(7) - s1, 2)))- dati.get(8)) <
          sqrt(abs(Math.sqrt(Math.pow(dati.get(6) - x2, 2)
          +Math.pow(dati.get(7) - s2, 2))) - dati.get(8))) {
      soluzione.add(x1 * 0.000001 + val.get(0));
      soluzione.add(s1 * 0.000001 + val.get(1));
   } else {

      soluzione.add(x2 * 0.000001 + val.get(0));
      soluzione.add(s2 * 0.000001 + val.get(1));
   }
}
```

Nel caso in cui il delta non sia uguale a 0 ma maggiore devo esaminare il valore di check. Esso può valere 0,1 o 2 a seconda delle circonferenze che sto ponendo a sistema (1 e 2 , 1 e 3 oppure 2 e 3). Nel listato 9 viene esaminato il primo caso. Subito dopo il controllo del check si passa a calcolare la distanza del punto p3 dai punti di intersezione di coordinate (x1,y1) , (x2,y2). Se l'algoritmo fosse esatto, la distanza da uno dei due punti di intersezione dovrebbe essere uguale al valore del raggio di p3. Non essendo così, ci si limita a sottrarre alla distanze calcolate la distanza dal router di p3 e a prendere il risultato che in valore assoluto è minore .

Listing 10: Delta minore di 0

```
// se delta è minore di 0 aumento il raggio delle 3 circonferenze
    cercando di farle incontrare
          // e scelgo come lanciare l'algoritmo. nell'ordine 1 e 2
              1 e 3 2 e 3

          if (d < 0) {
              if(iterazioni<50) {
                  iterazioni++;
                  if (check == 0) {
                      routerMap(dati, 1,iterazioni);
                  } else if (check == 1) {
                      routerMap(dati, 2,iterazioni);
                  } else if (check == 2) {
                      dati.set(4, dati.get(4) + 2);
                      dati.set(5, dati.get(5) + 2);
                      dati.set(8, dati.get(8) + 2);
                      routerMap(dati, 0,iterazioni);
                  }
              }else{
                  soluzione.add(dati.get(0));
                  soluzione.add(dati.get(1));
                  return soluzione;
              }
          }
      }

      return soluzione;
```

Infine è importante evidenziare che, qualora il delta fosse minore di 0, le circonferenze non si incontrerebbero mai. Si ripete per tanto la risoluzione del sistema cambiando le circonferenze e aumentando i raggi di 2 metri . Le ricorsioni possibili sono cinquanta ma generalmente dopo 10 iterazioni il sistema ammette soluzioni.
Le coordinate ottenute corrispondono alla posizione del router sulla terra in termini di latitudine e longitudine.
Esse andranno inserite all'interno della mappa per essere visualizzate. Si è

scelto, per comodità e disponibilità di documentazioni di usare le librerie di GoolgeMaps.

## 3.3 GoogleMaps

Google mette a disposizione degli sviluppatori un vasto pacchetto di servizi che consente di integrare al meglio le funzionalità delle mappe con il proprio applicativo. Nel caso di APostrophe è stato necessario utilizzare esclusivamente il servizio di localizzazione.
Dopo aver settato i parametri della mappa (visualizzazione , zoom iniziale...) , è importante settare i parametri di localizzazione che il GPS utilizzerà per tracciare la posizione dell'utente.

Listing 11: Parametri di localizzazione

```
mLocationRequest = new LocationRequest();
mLocationRequest.setInterval(5000);
mLocationRequest.setFastestInterval(2500);
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

Nel campionamento della posizione, sono numerose le possibilità di azione da parte dello sviluppatore in relazione alle esigenze. Come prima cosa, è importante settare l'intervallo di campionamento massimo, ossia il tempo massimo che intercorre tra una rilevazione ed un'altra, l'intervallo più rapido e l'accuratezza della rilevazione. Quest'ultima , può essere considerata come il dato più importante da studiare, poichè, designa il grado di precisione della rilevazione. Google, in questo caso, offre quattro opzioni che delineano un *trade-off* ben preciso tra consumi ed errore nel campionamento.
Naturalmente, ad un consumo più basso corrisponderà un errore maggiore. Come si nota nel listato 11 si è optato per il livello di accuratezza massimo, corrispondente ad un errore di pochi metri, al fine di poter utilizzare l'algoritmo di posizionamento nella maniera più efficace possibile e, al tempo stesso, garantire all'utente una facilità di connessione ai dispositivi che effettivamente si trovano vicini a loro.

All'interno dei metodi dell'activity che implementa le mappe, è gestita la raccolta delle coordinate e la chiamata al metodo di posizionamento: per il primo scopo, è stato istanziato un arraylist di "coordinate" che avrà dimensione massima 3. Ad ogni campionamento del GPS, le coordinate raccolte dopo il terzo campionamento, vengono sostituite con la politica del *first in first out* in modo tale da mantenere sempre il numero di rilevamenti in memoria costante. Nel metodo che si occupa di gestire tale operazione, è anche richiamato il servizio di ricerca delle reti WiFi captate in quel determinato punto.

Listing 12: Metodo di cambio della locazione

```
// quando cambia la posizione salvo le coordinate e le aggiungo
    all'arraylist
// contenente SOLO le ultime 3 rilevazioni
    @Override
    public void onLocationChanged(Location location) {

        if (tipo.equals("2")) {
          mLastLocation = location;
            if (mCurrLocationMarker != null) {
                mCurrLocationMarker.remove();
            }

            Coordinate nuovaCord;
            nuovaCord = new Coordinate(location.getLatitude(),
                location.getLongitude());
            coordinate.add(nuovaCord);

            if (coordinate.size() > 3) {
                coordinate.remove(0);
            }

// avvio e stoppo il servizio per rilevare i nuovi router alla
    data coordinata

            Intent i = new Intent(getApplicationContext(),
```

```
            WiFiService.class);
        startService(i);
        stopService(i);
    }
}
```

Come illustrato nel listato 12 il metodo di locazione *OnLocationChanged* permette di ricavare le coordinate di posizionamento dell'utente. Ad ogni cambio di locazione il metodo viene invocato e ripete l'operazione. Tale cambio, come spiegato in precedenza, avviene in un intervallo di 2,5 - 5 secondi.
Quando il servizio lanciato invia al *broadcast receiver* i dati inerenti ai wifi rilevati, avvengono una serie di controlli per verificare che della rete che si vuole posizionare siano presenti almeno tre rilevamenti, in caso positivo il dispositivo viene posizionato sulla mappa secondo il metodo descritto nella sottosezione precedente.

## 3.4 Servizi e Chrome Custom Tabs

L'utente, adesso, ha registrato un dispositivo sulla rete e lo ha reso raggiungibile e utilizzabile da tutta la community. Il passo successivo è quello di usufruire dei servizi di APostrophe, pertanto, accedendo alla mappa di visualizzazione dalla Homepage, si verrà immediatamente localizzati e appariranno tutti gli hot spot più vicini.
Una volta scelto quello preferito, è possibile stabilire una connessione a pagamento o gratuita per un tempo prestabilito. Appena si comincia a navigare, viene lanciata l'activity che permette di selezionare il servizio più utile all'utente in quel determinato momento. Allo stato attuale dello sviluppo, servizi utilizzabili sono solo 4 : Facebook , Instagram, YouTube e Maps. La scelta non è casuale , poichè, quando si è deciso di limitare l'utilizzo della connessione solo all'interno di APostrophe, si è anche deciso di limitare i servizi in modo da poter evitare attività illecite da parte degli utilizzatori. Per avere il controllo degli accessi e dei servizi sono state utilizzate le **Chrome Custom Tabs**.[14]

Le Chrome Custom Tabs sono nate per sopperire al bisogno degli sviluppatori di un sistema che permettesse di gestire gli URL all'interno della propria applicazione e senza il bisogno di implementare un browser customizzato sfruttando le *web view*.
Oltre ad un caricamento pagina decisamente più veloce , esse permettono la modifica della toolbar , l'inserimento di animazioni e la creazione di azioni personalizzate da inserire all'interno della toolbar.
Dal punto di vista della sicurezza si dimostrano molto valide poichè tramite l'utilizzo dei servizi di Google's Safe Browsing proteggono lo smartphone da siti potenzialmente nocivi. Le Chrome Custom Tabs offrono anche una importante funzione : tramite un sistema di callback permettono di gestire gli stati delle tabs in maniera asincrona così da avere un collegamento solido e continuo tra l'interfaccia di navigazione dell'utente e l'activity tramite la quale è stata chiamata.

Listing 13: Invocazione di una CCT

```
CustomTabsIntent.Builder builder = new
    CustomTabsIntent.Builder(getSession());

builder.setToolbarColor(Color.parseColor(TOOLBAR_COLOR)).setShowTitle(true);

builder.setStartAnimations(this, R.anim.slide_in_right,
    R.anim.slide_out_left);

builder.setExitAnimations(this, R.anim.slide_in_left,
    R.anim.slide_out_right);

builder.setCloseButtonIcon(BitmapFactory.decodeResource(getResources(),
    R.drawable.back));

CustomTabsIntent customTabsIntent = builder.build();
CustomTabsHelper.addKeepAliveExtra(this, customTabsIntent.intent);

if (viewId == R.id.facebook) {
   customTabsIntent.launchUrl(this,
       Uri.parse("https://facebook.com"));
```

```
}
```

Dopo aver settato le impostazione per la visualizzazione della toolbar e quelle di comparsa della tab, basta specificare l'url presso cui è situata la risorsa che si vuole ottenere e sarà già possibile navigare presso quell'indirizzo Si presti attenzione al metodo nel listato 13 *addKeepAliveExtra* necessario per la gestione delle **callBack**.

### 3.4.1 Gestione delle Callback

Le callback sono molto importanti per APostrophe: esse consentono di capire quando la Custom Tab è stata nascosta , in questo modo , si può facilmente gestire il sistema di sicurezza legata ai dispositivi, preservando la password della rete WiFi a cui si è connessi. Gli stati possibili di una tab sono i seguenti:

- NAVIGATION STARTED (1): mostrato quando la tab comincia a caricare una pagina;
- NAVIGATION FINISHED (2): mostrato quando la tab ha caricato la pagina;
- NAVIGATION FAILED (3): mostrato quando la tab non è riuscita a concludere un caricamento per via di un errore;
- NAVIGATION ABORTED (4): mostrato quando l'utente interrompe il caricamento completo di una pagina per esempio cliccando su un link;
- TAB SHOWN (5): mostrato quando la tab diventa visibile;
- TAB HIDDEN (6): mostrato quando la tab viene nascosta o chiusa;

Ogni volta che avviene un cambio di stato la finestra di navigazione invia un intent che può essere gestito.Nel caso di APostrophe è sufficiente considerare il sesto stato, ovvero quando la tab viene nascosta o chiusa.
Il problema della gestione del TAB HIDDEN è che rilascia lo stesso codice

di stato sia che venga chiusa, con conseguente ritorno alla scelta dei servizi, sia che venga nascosta e solo in quest'ultimo caso è necessario dimenticare la rete cui si è connessi.

Listing 14: Gestione di una CallBack

```
@Override
public void onNavigationEvent(int navigationEvent, Bundle extras) {

if(navigationEvent==6){

    try {
      TimeUnit.SECONDS.sleep(2);
   } catch (InterruptedException e) {
   e.printStackTrace();
   }
   if(riparte==true){
      riparte=false;
   }else{
      countDownTimer.cancel();
      dimentica();
      isDown=true;
   }

}
           }
```

Come è possibile osservare nel listato 14, è stato utilizzato un timer di attesa per capire se la tab è stata chiusa o nascosta: nel caso di chiusura, infatti si dovrebbe tornare all'activity servizi, quindi, se viene eseguito l' *onRestart()*, allora è bene mantenere la connessione, altrimenti si blocca il timer di utilizzo dei servizi e si dimentica la rete.

### 3.4.2 Timer e ripristino di sessione

Il timer visibile nell'interfaccia servizi è funzionale ad indicare il tempo ancora disponibile per l'utilizzo della connessione, è necessario che esso venga

bloccato e fatto ripartire a seconda degli stati dell'activity servizi.
Gli scenari che si potrebbero presentare sono molteplici: ad esempio, l'utente potrebbe ricevere una telefonata nella schermata servizi e non avere aperto nessuna tab, in questo caso sarà necessario bloccare il timer, eliminare la rete e , una volta terminata la telefonata permettere il ripristino della sessione. O magari è stata aperta una Chrome Custom Tab e nascosta per più di 2 secondi.
Per gestire tutte queste situazioni sono stati utilizzati molto,oltre alle callback delle tab, anche i metodi di stato delle activity. Nel listato 15 si nota come, tramite l'utilizzo di variabili boolean si possa gestire la connessione. Se l'activity sta ripartendo e non ho aperto nessuna tab allora sicuramente la connessione sarà stata persa, poichè è stata messa in pausa l'activity senza un motivazione giustificata. In questo caso viene visualizzato un messaggio di errore che consente di ritentare la connessione avendo salvato i dati necessari.
Le modalità di connessione sono del tutto analoghe a quelle osservabili nel listato 6.

Listing 15: Metodo di *onRestart()* dell'activity servizi

```
@Override
 protected void onRestart() {
    riparte=true;
     if(onCustomTab==false){
         dimentica();
         AlertDialog.Builder builder;
         if (Build.VERSION.SDK_INT >=
             Build.VERSION_CODES.LOLLIPOP) {
             builder =new AlertDialog.Builder(new
                 ContextThemeWrapper(Servizi.this,
                 R.style.AlertDialogCustom));
         }
         else {
             builder = new AlertDialog.Builder(Servizi.this);
         }
         countDownTimer.cancel();
```

```
            builder.setTitle("Attenzione!")
            .setMessage("Connsessione persa, hai ancora a
                disposizione:"+ min +" min e " + sec + " sec" )
            .setPositiveButton(android.R.string.yes, new
                DialogInterface.OnClickListener() {
               public void onClick(DialogInterface dialog, int
                   which) {
                           k = new Intent(getApplicationContext(),
                               WiFiConnect.class);
                           k.putExtra("bssid",bssid);
                           k.putExtra("key","servizi");
                           startService(k);
                           progress.show();
                       }
                    })
                    .show();
        }
        super.onRestart();
    }
```

## 3.5 APostrophe e l'interazione con l'utente

In APostrophe è stato progettato anche un sistema di personalizzazione delle informazioni e dei dispositivi di ogni utente. Ad oggi, è possibile modificare username, password ed email per ogni utente e modificare SSID e Password di ogni router. Quest'ultima caratteristica potrebbe essere utile qualora si desideri mantenere la password di default della rete e non quella suggerita dall'applicazione per la registrazione. In questo caso è sufficiente, dopo aver registrato l'hot spot, ripristinare la vecchia password e tramite il pannello di modifica , cambiare la password anche su APostrophe.
Questa azione, ma non è l'unica , non è sempre eseguibile : vi sono determinate funzioni utilizzabili solo quando vi è connettività alla rete.

### 3.5.1 Servizi online ed offline

Per problemi di concorrenza, alcuni servizi sono utilizzabili soltanto tramite connettività: infatti quando vengono modificate le informazioni personali oppure viene eliminato un router, l'operazione avviene immediatamente sul server e non sul dispositivo, così da essere notificato a tutta la community. Sono disponibili solo online:

- modifica dati personali dell'utente;
- modifica SSID e Password di ogni router;
- eliminazione dispositivi;
- ricarica saldo utente;
- aggiornamento mappe;

D'altro canto non era possibile per un servizio che offre connessione ad internet, non funzionare offline.
Le mappe, infatti, sono sempre consultabili in qualsiasi momento, per permettere all'utente di connettersi. Quando viene selezionata una rete che necessita di un pagamento e non si dispone di una connessione a dati mobili per verificare la bontà del pagamento, APostrophe stabilisce la connessione con la rete WiFi desiderata, tenta di effettuare il pagamento e in caso negativo disconnette immediatamente l'utente.

### 3.5.2 Gestione del database

Qualora in un futuro, gli utenti utilizzatori di APostrophe cominciassero ad essere considerevoli, i dispositivi registrati aumenterebbero in maniera esponenziale.
Pensare, quindi, di dover scaricare milioni e milioni di informazioni ogni qual volta si desideri aggiornare il database sembra quasi utopico. Per ovviare al problema si è pensato di implementare un servizio di gestione degli aggiornamenti.
Appena l'app viene installata su un dispositivo, viene settato tramite l'utilizzo di una SharedPreferences una "data x". Ogni qual volta si accede alla

home, l'applicazione tenta di scaricare dal server tutti quei dispositivi che hanno come data di aggiornamento una data più recente della data x.
In caso di successo dell'operazione la data corrente viene settata come nuova data x .

### 3.5.3 PayPal API

Per invogliare gli utenti a condividere i propri dispositivi, è stata introdotta la possibilità di stabilire un costo al minuto che un ipotetico utilizzatore dovrà pagare qualora voglia usufruire della connessione. Ogni utente , quindi, dispone di un conto collegato all'account, tramite il quale verranno effettuate sia transazioni in entrata e sia transazioni in uscita.
Per poter cominciare ad utilizzare reti a pagamento, è necessario che tale conto venga ricaricato, per questo si è scelto di utilizzare i servizi e le API di PayPal [15] con lo scopo, solo momentaneo di simulare un pagamento verso i gestori dell'applicazione: le ricariche possibili saranno di 5 o 10 euro.
Facilmente rintracciabili tramite il sito ufficiale, i numerosi servizi messi a disposizione consentono all'utente di incorporare i metodi di pagamento PayPal alla propria App o Web-App permettendo di svolgere o simulare transazioni.
La PayPal Sandbox è un ambiente di testing per i prodotti PayPal interamente autogestito e facente parte del pacchetto API Basics. Provvede alla creazione di un spazio protetto in cui poter simulare tutte le attività inerenti al mondo PayPal senza che nessun vero account venga intaccato.
Per essere utilizzata richiede che lo sviluppatore sia registrato presso PayPal Developer e che abbia quindi un conto, anche senza denaro, a suo nome. Non serve altro che dichiarare la nuova applicazione sul sito degli sviluppatori per ricevere il PayPalClientId necessario ad avviare l'activity di pagamento.
Una volta ottenute le credenziali, sarà sufficiente importare la libreria necessaria e configurare l'ambiente, richiedendo una SandBox con il proprio codice cliente. Quando l'utente cliccherà sul bottone di ricarica l'activity partirà e gestirà automaticamente tutti i passaggi che vanno dalla verifica delle credenziali di colui che deve pagare alla conferma di avvenuto paga-

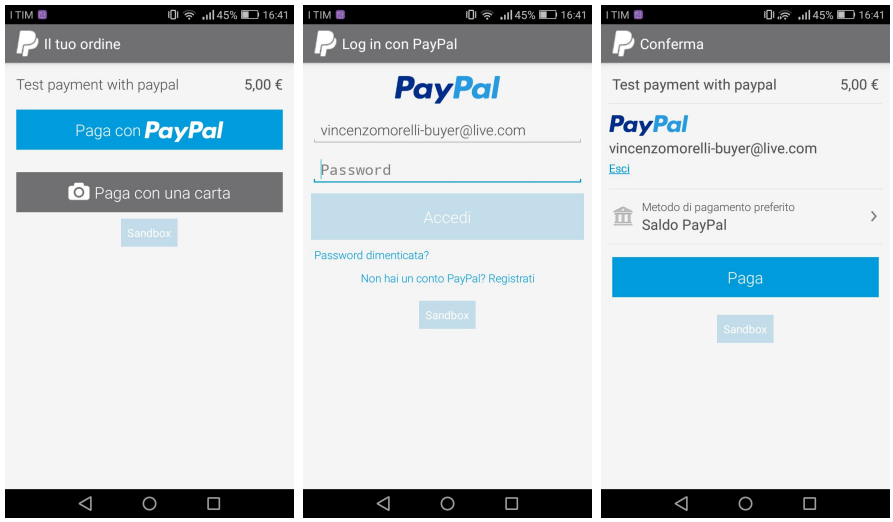


Figura 12: Pagamento con Sandbox PayPal

mento. Si parla dunque di una *ActivityOnResult* ovvero una activity che restituisce un risultato a runtime.

Listing 16: Setup dell'environment PayPal

```
 // configuro l'environment necessario per avviare paypal sandbox

m_configuration = new PayPalConfiguration()
.environment(PayPalConfiguration.ENVIRONMENT_SANDBOX)
.clientId(m_paypalClientId);

m_service= new Intent(this.getActivity(), PayPalService.class);

m_service.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION,m_configuration);

 if(v==payPal){

       payment=new PayPalPayment(new BigDecimal(5),"EUR","Test
           payment with paypal",PayPalPayment.PAYMENT_INTENT_SALE);

   Intent intent=new
       Intent(this.getActivity(),PaymentActivity.class);
        intent.putExtra(PayPalService.EXTRA_PAYPAL_CONFIGURATION,m_configuration);
        intent.putExtra(PaymentActivity.EXTRA_PAYMENT,payment);
        startActivityForResult(intent,m_paypalRequestCode);
 }
```

Appena si è conclusa l'operazione di pagamento, a seconda dell'esito viene richiesto lo stato del pagamento all'API .
È adesso possibile elaborare il risultato e ricaricare il conto in caso di pagamento approvato o visualizzare un errore in caso di pagamento fallito.

## 3.6 Comunicazione Client-Server e creazione database MySQL-SQLite

Dalla registrazione alla gestione del pagamento, dal posizionamento alla modifica dei dati dell'utente, quasi tutte le operazioni se non proprio tutte vengono svolte sfruttando l'architettura client server ed i database dell'applicazione. Per effettuare le chiamate ci si è avvalsi di un pacchetto di classi già presente su GitHub, tuttavia, l'elaborazione dati e la sezione relativa al server risultano del tutto nuove.

### 3.6.1 Comunicazione Client-Server: lato Client

Dal dispositivo vengono effettuate numerose chiamate al server sfruttando una architettura ben precisa : i metodi utilizzati sono metodi di GET o di POST a seconda che si voglia ottenere o modificare una risorsa. Le chiamate possono essere distinte in macro sezioni : quelle che coinvolgono il database locale e quelle che non lo coinvolgono . Un'altra distinzione si basa su chiamate che in risposta ricevono parametri e chiamate che ricevono solo conferma di successo o fallimento dell'operazione. Ad esempio, per la visualizzazione dei dati dell'utente, il client invia come parametro l'username e riceve in risposta anche la password e l'email. La richiesta di modifica dei dati, invece, ritorna soltanto successo o fallimento. La comunicazione più particolare, però, è quella che riguarda il download dei nuovi dispositivi. Essa, oltre ad utilizzare il sistema di gestione dell'ora di aggiornamento descritto nella sottosezione 3.5.2, necessita anche del passaggio del contesto e dell'activity per poter interagire con il database SQLite .

Listing 17: Configurazione chiamata *syncDB()*

```
prefs= getSharedPreferences("dbsync", MODE_PRIVATE);
editordb= getSharedPreferences("dbsync", MODE_PRIVATE).edit();
SyncServer get=new SyncServer();
get.setDate(prefs.getString("syncDate","2017-07-31 15:02:19"));

//setto il context e registro l'activity che implementa
    l'interfaccia per gestire la callback

get.setCallback(this);
get.setContext(this,this);
try {
    //chiamo la sincronizzazione del database
    get.syncDB()
 }
 catch (Exception e) {
    e.printStackTrace();
}
```

Essendo ,la comunicazione client-server, asincrona non è possibile gestire le risposte all'interno dell'activity se non tramite l'implementazione di un'interfaccia. Come mostrato nel listato 17, serve che l'activity si identifichi presso l'interfaccia e che implementi (non mostrato in figura) i metodi di onSuccess() e onFaliure() che verranno fatti partire dalla classe che gestisce la chiamata.
Quando si effettua la chiamata vera e propria è necessario settare il metodo HTTP che si intende utilizzare in base all'utilizzo che si deve fare della risorsa, definire l'url al quale si effettuerà la richiesta e passare i parametri per effettuarla. Quest'ultimo step avviene tramite l'utilizzo di un' *Hash Map* dove si specifica il nome del parametro (chiave) e il valore che andrà letto lato server (valore).

Listing 18: Creazione e codifica parametri

```
HttpCall httpCall = new HttpCall();
httpCall.setMethodtype(HttpCall.GET);
httpCall.setUrl("http://apostrophe.altervista.org/server.php");
HashMap<String,String> params = new HashMap<>();
String encMetodo = MCrypt.bytesToHex( mcrypt.encrypt("sync") );
String encDate = MCrypt.bytesToHex( mcrypt.encrypt(date) );
params.put("metodo",encMetodo);
params.put("data",encDate);
httpCall.setParams(params);
```

Dopo aver settato tutti i parametri come nel listato 18, si procede alla richiesta HTTP, la quale ritorna una risposta di tipo stringa. Tale stringa, viene poi rielaborata e considerata , come si vede nel listato 19, come un array di oggetti JSON procedendo in caso di successo alla notifica presso il chiamante che ne implementa l'interfaccia.

Listing 19: Richiesta HTTP

```
new HttpRequest() {

    @Override
     public void onResponse(String response) throws Exception {
```

```
   super.onResponse(response);
   if (response.equals("error")) {
      interf.onError(0);
      return;
   }

    String decrypted = new String( mcrypt.decrypt(
        response.replace(" ","") ) );

    JSONObject jsonObj = new JSONObject(decrypted);
    JSONArray contacts = jsonObj.getJSONArray("router");

    for (int i = 0; i < contacts.length(); i++) {

   JSONObject c = contacts.getJSONObject(i);

   String id=c.getString("ID");
   String SSID = c.getString("SSID");
   String BSSID = c.getString("BSSID");
   Double lat = c.getDouble("latitudine");
   Double lon= c.getDouble("longitudine");
   String proprietario = c.getString("proprietario");
   String stato = c.getString("stato");
   String password= c.getString("password");
   String data=c.getString("data");
   String tariffa=c.getString("tariffa");

db.insertDB(attivita,SSID,BSSID,lat,lon,stato,proprietario,password,data,tariffa);

    }

     interf.onSuccess(0);

    }

}.execute(httpCall);
```

Dopo aver ricavato le singole componenti dei dati, è possibile, nel caso del

metodo *syncDB()*, la memorizzazione dei router su database SQLite.
Come ampiamente spiegato in precedenza, non tutti i dispositivi vengono scaricati, ma solo quelli più recenti dell'ultimo download effettuato. Singolare è la procedura di eliminazione : infatti, i dispositivi vengono eliminati solo su SQLite , quindi in locale e non sul server, poichè non ci sarebbe modo di notificare a tutti gli altri dispositivi dell'avvenuta eliminazione.
A questo scopo si è deciso di considerare lo stato del router come sistema per la rimozione dai database locali.
Quando un dispositivo viene eliminato, sul server, il suo stato viene modificato da disponibile a cancellato e anche il campo data viene aggiornato. Quando uno smartphone che aveva il dispositivo, ora eliminato, tra quelli memorizzati, scarica i nuovi aggiornamenti, prima di inserire i nuovi hot spot nel database verifica che il loro stato non risulti cancellato , in caso contrario effettua l'operazione di cancellazione.
La progettazione del database SQLite è in tutto e per tutto simile a quella di un normale database MySQL.
Android e Java mettono a disposizione la classe *SQLiteOpenHelper* che consente la creazione del database in maniera rapida ed efficiente. Le tabelle, così come in MySQL, vengono create tramite query. Nelle operazioni di insert sono utilizzati i *content values* per l'attribuzione delle coppie colonna valore , le select sfruttano, invece un cursore per scorrere i risultati e prendere solo i campi di interesse.

Listing 20: Esempio di select in SQLite

```
String selectQuery;

        selectQuery = "SELECT * FROM "+ TABLE_ROUTER +" WHERE
            BSSID='"+COL_BSSID+"'";
        SQLiteDatabase db = this.getReadableDatabase();
        try{
            Cursor cursor    = db.rawQuery(selectQuery, null);
            String[] data    = null;
            if (cursor.moveToFirst()) {
                do {
                    res.add(cursor.getString(1));
                    res.add(cursor.getString(7));
```

```
            }
            while (cursor.moveToNext());
        }
        cursor.close();}catch (Error e){
        System.out.println(e);
    }
    return res;
```

Nel listato 20 , dalla query, vengono estratti soltanto il secondo ed l'ottavo campo (perchè il cursore parte da 0).

### 3.6.2 Comunicazione Client-Server: lato Server

Lato server, la struttura è decisamente meno complessa. Potendo utilizzare i tool di Altervista come **phpMyAdmin**, la creazione del database in rete è risultata estremamente semplice. Per quel che riguarda l'architettura di comunicazione con il client, si è deciso di adoperare php per compatibilità con i sistemi su cui la struttura è situata. Sono state create diverse pagine php a cui vengono fatte le chiamate, suddivise in base al metodo utilizzato oppure che si tratti di query verso una o l'altra tabella.
L'impostazione è sempre la stessa, ovvero , dopo aver effettuato la connessione al database tramite php, viene esaminato il **metodo** di chiamata , settato lato client. In base al metodo, viene eseguita la query richiesta ed inviato il risultato. Se quest'ultimo è una stringa viene inviato direttamente, altrimenti prima viene strutturato in un file **json**.

```
 if($metodo=='delete'){

$data= $mcrypt->decrypt($_POST['data']);
$bssid= $mcrypt->decrypt($_POST['bssid']);

$sql = "UPDATE posizione SET stato='cancellato',
    data='".$data."' WHERE BSSID='".$bssid."'";


 $result=$conn->query($sql);
```

```
 $conn->next_result();
}
```

## 3.7 Sicurezza e crittografia

In quest'ultima sezione, infine, vengono descritti tutti i metodi utilizzati per preservare i dati dell'utente e garantire la corretta protezione nelle diverse sezioni dell'APP.
Dopo che, in fase di progettazione, sono state individuate tre diverse zone critiche in cui potesse avvenire un attacco informatico nel tentativo di sottrarre informazioni protette ad APostrophe, si è deciso di gestire momentaneamente la protezione dei dati solo durante il trasferimento da client a server.
Obbiettivo futuro importante è garantire che ogni sistema sia sicuro sotto tutti i punti di vista. In seguito a diverse ricerche sul web, il metodo di cifratura più affidabile e allo stesso tempo semplice da implementare è risultato quello disponibile su *www.androidsnippets.com*.
Il codice si compone di due classi, una importabile in Java e una in php. L'algoritmo utilizzato è AES o Rijndael a 128bit. Dopo aver inserito le classi nel progetto è sufficiente invocare i metodi di crittazione e decrittazione sulle stringhe interessate per ottenere una protezione "sufficiente" durante il passaggio dei dati.
La scelta di utilizzare un algoritmo a chiave privata presenta naturalmente dei pro e dei contro.
Sicuramente è molto più veloce e semplice da implementare (non solo perchè sia Java che php posseggono già le librerie per cifrare) .Se si fosse scelto un algoritmo a chiave pubblica sarebbe stata necessaria la certificazione della chiave.
D'altro canto la chiave privata condivisa tra client e server, al momento è unica e non è stato ancora implementato un metodo per la creazione e la trasmissione di una chiave di sessione.

CAPITOLO 4

# *Conclusioni e sviluppi futuri*

# 4 Conclusioni e sviluppi futuri

L'obbiettivo della tesi è stato quello della creazione di una nuova WiFi community network che permettesse agli utenti di usufruire di un servizio di rete sempre stabile ed efficiente.
Al tempo stesso si è cercato di preservare il più possibile le informazioni private degli utenti e dei dispositivi.
Durante l'implementazione, le componenti più impegnative da realizzare sono state l'implementazione dell'algoritmo di posizionamento e i servizi di connessione WiFi.
Il comportamento "imprevedibile" del WiFi Manager ha reso faticosa l'implementazione di un algoritmo in grado di gestire gli stati della connessione.
La scelta (più semplice) di non utilizzare la posizione attuale dell'utente per sistemare il router sulla mappa, deriva invece da due fattori:

1. L'intensità del segnale al momento della mappatura;
2. Il possibile riutilizzo dell'algoritmo in future *features* aggiunte all'app.

Per analizzare il primo caso, si immagini che l'utente al momento del posizionamento, si trovi nel punto di massima distanza dal router. Il raggio d'azione indicato dall'app risulterebbe sbagliato e gli utenti, successivamente non sempre riuscirebbero a connettersi.
In futuro tale algoritmo potrebbe essere utilizzato, per esempio, per una mappatura in tempo reale dei dispositivi registrati al fine di collocare il dispositivo esattamente nel punto di massima intensità del segnale in quel determinato istante, tenendo conto dei segnali di interferenze che dal momento della mappatura, si sono interposti tra l'utilizzatore e l'hot spot.

Sempre dal punto di vista dell'efficienza del servizio, si potrebbe creare un sistema di diagnosi dell'intensità che aiuti ad individuare i punti di interferenza del segnale, i campi magnetici, fornendo all'utente un servizio che scelga per lui il dispositivo più adeguato a cui connettersi in quel determinato momento. Il sistema di pagamento potrebbe essere reso più efficiente: dalla ricarica, possibile solo se si è connessi ad una rete NON tramite l'applicativo, al sistema di pagamento.

In particolare, quest'ultimo, è disponibile ed utilizzabile in maniera mirata su una determinata rete. Sebbene una volta persa la connessione, sia possibile recuperare la sessione, molto spesso i "minuti di connessione" acquistati non vengono utilizzati completamente.
Molto utile sarebbe l'implementazione di un sistema che permetta la gestione della connettività dinamica. Inserita la fascia di prezzo massimo utilizzabile, l'applicativo sceglierebbe automaticamente la connessione da stabilire in modo tale da potersi spostare e contemporaneamente continuare a navigare.

Per quel che concerne i servizi offerti invece, l'idea è quella di estenderli il più possibile compatibilmente con la tutela dell'utente che sceglie di condividere il punto di accesso ad internet.
Sicuramente le Chrome Custom Tabs rendono facile e rapida la navigazione, tuttavia, il servizio è ristretto al loro utilizzo esclusivo per motivi di sicurezza. Offrire una navigazione a 360 gradi che consenta di utilizzare anche altri applicativi con le activity di APostrophe in background, sarebbe senza dubbio un traguardo importante, anche se, tale operazione comporterebbe ad una riprogrammazione delle api del WiFi Manager che risulterebbe dispendiosa in termini di risorse, oltreché impegnativa.
Infine, la gestione della sicurezza nei vari ambiti. Come già illustrato nei capitoli 2 e 3 è necessario che i dati vengano protextit in tutte e tre le fasi di transizione : sul server, sul client e nel passaggio.

Seppur in secondo piano, un ruolo importante ha avuto anche l'aspetto grafico. Non possedendo già tutte le competenze necessarie alla creazione di un applicativo, è stato necessario impegnarsi non poco nella realizzazione delle interfacce sfruttando i layout oppure la creazione dei menu tramite Custom List Adapter o ancora la creazione di Custom InfoWindow all'interno per le mappe.

Concludendo, questo lavoro di tesi ha una duplice funzione : l'approfondimento delle tecnologie utilizzate su ogni fronte, dagli aspetti grafici, all'utilizzo di servizi ed API a me prima sconosciute ma più di tutti ho

avuto l'opportunità di implementare la prima applicazione nata da una idea personale, seppur non rivoluzionaria, ma con importanti componenti innovative.

# *Ringraziamenti*

*In primo luogo, vorrei ringraziare il professor Luciano Bononi ed il dottor Luca Bedogni per la serietà e l'impegno con cui mi hanno seguito durante tutto il lavoro di tesi, dalla progettazione alla stesura di questo elaborato. Sempre disponibili sono riusciti, con preziosi suggerimenti, a consigliarmi nelle scelte più difficili .*

*Grazie ad Antonio per il logo e grazie a tutti i miei amici, da quelli storici a quelli conosciuti in questi ultimi tre anni, per avermi consigliato, aiutato e sopportato in tutte le difficoltà, per aver partecipato con entusiasmo alla mia vita universitaria e per i consigli legati anche allo sviluppo di Apostrofe.*

*Un ringraziamento speciale va, soprattutto, ai miei genitori e a mia sorella, per avermi concesso l'opportunità di studiare lontano da casa e per avermi trasmesso i valori di vita necessari ad affrontare questo percorso triennale con passione e positività.*

# Riferimenti bibliografici


[1] *G. Camponovo and A. Picco-Schwendener, "Motivations of Hybrid Wireless Community Participants: A Qualitative Analysis of Swiss FON Members," 2011 10th International Conference on Mobile Business, Como, 2011, pp. 253-262.*

[2] *Fon WiFi community Network. https://fon.com/wifi-community-network/.*

[3] *Jose-Aurelio Medina Garrido, Salustiano Martinez-Fierro. (2008). Cases on Information Technology Entrepreneurship*

[4] *Wiman. https://www.wiman.me/it*

[5] *WiFi Italia IT. http://wifi.italia.it/it/.*

[6] *A.Longo "Il governo liberalizza il Wi-Fi, Decreto Fare cambiato in extremis." Il Sole 24 ore.*

[7] *T. Eisner, D. Lütke-Wiesmann and B. Steersman, "Protecting user privacy in WiFi sharing networks," 2010 IEEE Globecom Workshops, Miami, FL, 2010, pp. 1979-1983.*

[8] *C.Frediani . "Il Parlamento vuole allungare a 6 anni la conservazione di tabulati e dati internet." La Stampa.*

[9] *ERDPlus E-R Diagrams. https://erdplus.com/*

[10] *P. Buonavoglia. "La Crittografia da Atbash ad RSA". http://www.crittologia.eu/*

[11] *G. l. Guo, Q. Qian and R. Zhang, "Different Implementations of AES Cryptographic Algorithm," 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, 2015, pp. 1848-1853.*

*[12] Android SharedPreferences. https://developer.android.com/.*

*[13] Android WiFi Manager APIs. https://developer.android.com/.*

*[14] Chrome Custom Tabs. https://developer.chrome.com/.*

*[15] PayPal APIs. https://developer.paypal.com/*