

# **UNIVERSITA' TELEMATICA “e-Campus”**

**FACOLTA' DI INGEGNERIA  
CORSO DI LAUREA IN INGEGNERIA INFORMATICA E  
DELL'AUTOMAZIONE**

---

---

## **SVILUPPO DI UNA APPLICAZIONE ANDROID DI STATISTICA PER IL MERCATO IMMOBILIARE**

*Relatore:*

**Chiar.mo Prof.re Alessio MERLO**

*Tesi di Laurea di:  
Donato PEPE*

*Matricola numero 001014428*

---

**ANNO ACCADEMICO 2013-2014**

## AUTORIZZAZIONE ALLA CONSULTAZIONE DELLA TESI DI LAUREA

Il sottoscritto DONATO PEPE  
N° di matricola 001014428 nato a TARANTO il 24/02/1977  
autore della tesi dal titolo Sviluppo di una applicazione ANDROID  
DI STATISTICA PER IL MERCATO IMMOBILIARE

- AUTORIZZA  
 NON AUTORIZZA

la consultazione della tesi stessa, fatto divieto di riprodurre, parzialmente o integralmente, il contenuto.

Dichiara inoltre di:

- AUTORIZZARE  
 NON AUTORIZZARE

per quanto necessita l'università telematica e-Campus, ai sensi della legge n. 196/2003, al trattamento, comunicazione, diffusione e pubblicazione in Italia e all'estero dei propri dati personali per le finalità ed entro i limiti illustrati dalla legge.

Data 29/05/2016

Firma Donato

## **INDICE**

INTRODUZIONE

COSA SERVE PER IL SOFTWARE

SPECIFICHE TECNICHE

PROBLEMATICA NELLO SVILUPPO SOFTWARE

1.	MODALITA' DI PRIMO ACCESSO .....	11
1.1	INTRODUZIONE .....	11
1.2	CREAZIONE PROFILO UTENTE .....	12
2.	INSERIMENTO DI UN IMMOBILE .....	16
2.1	GENERALITA' .....	16
2.2	COME INSERIRE UN IMMOBILE .....	16
2.2.1	Ricerca indirizzo .....	16
2.2.2	Inserimento caratteristiche dell'immobile .....	19
2.3	COME RICHIAMARE UN IMMOBILE SALVATO .....	26
2.4	SALVATAGGIO IMMOBILE NEL DATABASE .....	27
3.	ANALISI STATISTICA DEI DATI IMMOBILIARI .....	30
3.1	GENERALITA' .....	30
3.2	FILTRO DATI .....	30
3.3	MODALITA' DI CONTATTO UTENTI .....	32
3.4	ANALISI STATISTICA DEI DATI .....	33
3.4.1	Visualizzazione tabelle .....	34
3.4.2	Visualizzazione grafico .....	35
4.	ALLEGATO: SORGENTI SVILUPPATI NELLA TESI .....	37

BIBLIOGRAFIA

## INTRODUZIONE

Di seguito, prima di illustrare la guida all'uso del software *Open Realtor*, si espongono i possibili utenti, usi per cui il software è stato realizzato e una panoramica sulle potenzialità del programma.

La crisi immobiliare, iniziata approssimativamente a metà anno 2006, ha colpito duramente il mercato rallentando la vivacità di acquisto e/o affitto degli immobili che si viveva prima. In questo clima di sfiducia e di molti ripensamenti è nata l'idea di sviluppare un software *Open Realtor* per statistiche immobiliari su device mobile. Nel mercato esistono già software per PC e per mobile nella gestione degli immobili, rivolta a professionisti del mercato immobiliare, ma la realizzazione del programma *Open Realtor* ha la presunzione di voler raggiungere tre obiettivi:

- permettere l'accesso al sistema indipendentemente dal mobile utilizzato;
- consentire sia una valutazione degli andamenti dei prezzi di vendita degli immobili in Italia e nel mondo sia una loro valutazione temporale attraverso analisi statistiche;
- rendere internazionale la visura degli immobili.

Poiché *Open Realtor* si rivolge a tutto il pubblico, il raggiungimento di tali obiettivi avviene con semplici passi, spiegati con maggior dettaglio nei capitoli successivi, grazie all'ausilio di un'interfaccia grafica molto semplice e intuitiva, che risulta di facile comprensione sia per il professionista, che può avere dimestichezza nel settore, sia per un utente generico. Scegliere il sistema *Open Realtor*, per inserire e/o consultare i dati del proprio immobile o di altri, trova la sua convenienza anche dal punto di vista economico, poiché a oggi è free, solo la richiesta di poter essere messo in contatto con l'account di chi ha inserito l'immobile avviene tramite il software developer, che, nel

caso in cui avvenga una transazione, si assicura una provvigione mediante la stipula di un contratto.

Il raggiungimento del primo obiettivo, ovvero consentire l'accesso al sistema indipendentemente dal mobile utilizzato, è stato perseguito realizzando il programma con *Apache Cordova*, ovvero ex-*PhoneGap*, un framework di sviluppo mobile open source, con l'utilizzo delle seguenti tecnologie standard web HTML5, CSS3 e JavaScript per lo sviluppo sulle piattaforme *Android*, *BlackBerry*, *BlackBerry 10*, *iOS*, *Windows Phone*, *Windows 8* e *Tizen*, anche se al momento la sperimentazione è stata eseguita solo su *Android*. L'utilizzo di *Apache Cordova*, compilata su diversi device, permette di pubblicare la web app sui relativi App Store, dando così la possibilità di avvalersi del programma indipendentemente dalla scelta del mobile utilizzato.

L'installazione del programma *Open Realtor*, così come le altre applicazioni che si installano su mobile, richiede la registrazione al software con un'informazione minima di contatto. In tale procedura è indispensabile la scelta di un nickname e di un'e-mail valida per proseguire con la procedura di verifica mediante e-mail di convalida dell'account. Il software è stato realizzato pensando anche alla tutela della privacy, ovvero dall'App sono visibili solo i dati dell'immobile e non di chi li ha inseriti, dati che comunque rimangono immagazzinati nel database. Questa particolare attenzione è a garanzia dell'anonimato degli utenti che fruiscono del programma, i contatti tra gli utenti possono avvenire solo tramite il software developer nel caso in cui ci sia un interesse su un immobile inserito nel database. Inoltre per una maggiore tutela degli utenti, i dati inseriti non potranno essere manipolati da attacchi hacker, poiché il database centrale e l'interfaccia di comunicazione tra frontend, ovvero *Open Realtor*, e il server di backend

sono entrambi protetti da password differenti, la cui gestione compete solo al software developer.

Il raggiungimento del secondo obiettivo, ovvero la valutazione dei prezzi, si persegue popolando il database. Inseriti i dati personali e avuto accesso al software, l’utente per inserire i dati di un nuovo immobile deve procedere alla localizzazione dell’oggetto che può avvenire o inserendo direttamente l’indirizzo, qualora fosse noto, o attraverso il consulto delle mappe di *Google*. Grazie a questo semplicissimo step gli elementi dell’immobile possono essere georeferenziati<sup>1</sup>, ovvero un’icona sullo schermo della mappa aiuta a visualizzare l’informazione inserita con le caratteristiche peculiari di localizzazione geografica del dato, esattamente come *Google Maps*, in cui è possibile cercare negozi o località di interesse dei quali vengono fornite non solo le tipiche informazioni che restituisce un motore di ricerca, ma vengono anche evidenziate sulla mappa la posizione geografica ad essi riferita. Dopo la localizzazione si passa all’inserimento dei dati dell’immobile in oggetto, tale inserimento segue una procedura abbastanza intuitiva grazie ai selettori che sono di facile comprensione sia per il professionista sia per l’utente generico. Infatti, sullo schermo compaiono diversi selettori che se cliccati consentono ciascuno di aprire una tendina in cui appaiono una serie di voci, pertanto, basandosi sulle reali caratteristiche dell’oggetto che si sta inserendo, si passa alla scelta delle specifiche dell’immobile. L’utente a sua discrezione può archiviare nel proprio database fino a un massimo di dieci foto per ogni immobile inserito, pertanto si lascia la possibilità agli utenti di visualizzare foto d’interni ed esterni e, se disponibile, anche la planimetria. L’ottimizzazione del software, nel procedimento di archiviazione dati, è

---

<sup>1</sup> Per georeferenziazione s’intende la “Tecnica di attribuzione di coordinate geografiche a un oggetto grafico, usata nelle procedure di cartografia computerizzata e nella costruzione di basi cartografiche digitali.” (cit. Treccani, L’enciclopedia italiana)

stata realizzata dando la possibilità di inserire le specifiche in modalità offline, ovvero anche se l'operatore telefonico non dovesse consentire l'immediato accesso ad internet, l'utente può inviare al database le informazioni inserite in un altro momento.

L'utente anche senza account può accedere all'analisi statistica dei prezzi sugli immobili in una località, individuata l'area si visualizzano al massimo cinquanta schede presenti nel database, ordinate per data dal più recente, selezionabili per fare la valutazione e generare un grafico a interpolazione lineare che visualizza l'andamento dei prezzi.

Quanto esposto sino ad ora fa comprendere in che modo *Open Realtor* potrebbe diventare uno strumento utile sia al professionista, che opera nel campo degli immobili come imprese, Enti, Istituti Bancari e consulenti, sia al privato, perché facilita la scelta dell'indice di valutazione sugli immobili. Questa potenzialità aumenta in rapporto al numero d'informazioni che popolano il database, perché, definire tale indice, diventa più razionale se le statistiche sono calcolate con un campione numeroso e aggiornato, ovviamente restituiscono un risultato più preciso ed è per questo che si è scelto di rendere free l'applicazione. Detto questo, diventa anche facile comprendere la potenzialità del sistema nella gestione della valutazione sugli immobili, cui devono rispondere i professionisti in tale campo, che per il loro mestiere devono gestire numerosi dati e soprattutto rispondere velocemente alle richieste di mercato, fornendo ai loro clienti le informazioni più appropriate alle loro esigenze.

Il terzo obiettivo nell'uso di *Open Realtor* è di rendere internazionale la visura degli immobili, perché se il suo utilizzo trova grande applicazione per la valutazione degli immobili a maggior ragione diventa un ottimo strumento per deciderne l'acquisto, non solo in Italia ma nel mondo. L'utilizzo del software si presta per usi in paesi anche non

italiani, al momento è stata inserita oltre all’italiano la lingua inglese, perché è quella riconosciuta come internazionale, ma per scelte future il programma si presta all’inserimento di altre lingue. La preferenza della lingua non è un problema che l’utente deve risolvere perché la scelta è affidata al software, che riconosce in automatico la lingua da visualizzare in base a quella settata sul device. L’internazionalizzazione delle tabelle impiegate nel programma è garantita dal database MySQL del server centrale che utilizza di default, tra le possibili scelte, la codifica di caratteri UTF-8.

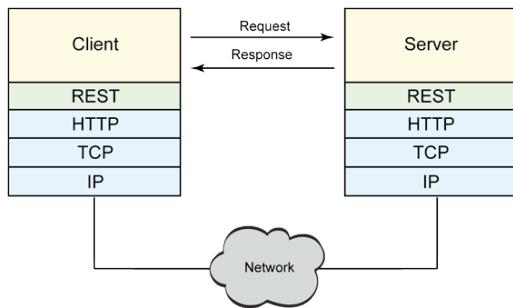
## COSA SERVE PER IL SOFTWARE

L’installazione e l’utilizzo di *Open Realtor* necessita di:

- device *Android* (smartphone e tablet);
- connessione ad internet per consultazione e aggiornamento dati.

## SPECIFICHE TECNICHE

Per gestire le funzionalità del server web è stata adottata l’architettura del sistema *RESTful* mediante il micro framework *Slim PHP*, con questo strumento sono stati messi a disposizione dei metodi con cui è possibile dare servizi web comunicanti con il protocollo HTTP. Il framework PHP SLIM supporta tutti i metodi necessari per una API REST, poiché, anche se si presenta leggero e pulito, fornisce un’architettura abbastanza solida nella gestione delle chiamate in ingresso dall’App *Open Realtor*. Per quanto detto, si è scelto un server web con le seguenti tecnologie: *Apache*, *PHP* e *MySQL*.



**Figura 1** Schema flusso informazioni

Il database *MySQL* è composto da una tabella utenti e da una tabella per ogni provincia contenente gli immobili inseriti dall’utente, che si vengono a generare automaticamente dal momento in cui è inserito l’immobile, se la tabella per quella provincia non dovesse esser stata già creata da un altro utente.

I nuovi inserimenti e aggiornamenti delle tabelle sono realizzati con i metodi REST, in particolare quelli utilizzati dal server backend sono:

- gestione dell’invio di e-mail: a disposizione dell’utente per chiedere o informazioni sul software o su una scheda immobiliare o per l’invio del codice di convalida dell’account, queste funzioni sono possibili poiché si utilizza l’interfaccia in PHP del server backend;
- tabella utenti: inserimento, aggiornamento e controllo esistenza di altri nickname uguali;
- tabelle schede immobiliari: inserimento e aggiornamento di una scheda se il proprietario corrisponde.

Le foto dell’immobile, che l’utente può gestire nelle schede immobiliari, non sono registrate come file nello spazio web, ma nel database per assicurarsi una sua migliore organizzazione e facile manutenzione. Inoltre, il software scala automaticamente la riso-

luzione delle immagini a un valore minimo fissato, evitando in tal modo di riempire velocemente la memoria del database.

Nell'App *Open Realtor* le chiamate ai metodi del server backend sono fatte mediante la programmazione in AJAX, a seconda delle funzionalità le chiamate sono asincrone, ad esempio invio immobili al database, o sincrone, ad esempio nel controllo del nickname. Utilizzando AJAX come linguaggio di programmazione, lo scambio dati nell'applicazione client-server è con JSON, utilizzato per la semplicità della sua struttura.

Nella fase di sviluppo è stato utilizzato il tool a linea di comando cURL, per effettuare i debug e i test del server backend; questo tool ha permesso di testare velocemente tutte le funzionalità messe a disposizione dal server backend per poi essere implementate nell'applicazione client.

Il calcolo della superficie equivalente<sup>2</sup> avviene nell'interfaccia PHP, in tal modo non si ha la necessità di aggiornare i software dei mobile per modifiche sui pesi dei dati inseriti, piuttosto è sufficiente agire sul server web. Tale valore viene calcolato dal programma quando l'utente inserisce o aggiorna i dati tecnici di un immobile.

Nell'App si sono utilizzate le librerie di *JQuery* e *JQuery mobile* per le loro peculiari caratteristiche. La prima libreria offre un'ampia gamma di funzionalità, quali la

---

<sup>2</sup> Per “superficie equivalente” di un immobile s’intende il valore di una superficie calpestabile corretta con i dati tecnici relativi sia ad aspetti quantitativi, ad esempio come il piano, il n. di bagni e così via, sia ad aspetti qualitativi, ad esempio come il paesaggio, i servizi e così via.

manipolazione degli stili CSS e degli elementi HTML, di effetti grafici, di facili metodi per chiamate AJAX cross-browser, di brevità del codice utilizzato, della possibilità di ricercare elementi e gestirli. La seconda libreria, estensione della prima, è progettata per poter realizzare un'unica applicazione fruibile da un largo numero di dispositivi smartphone, tablet e piattaforme desktop.

Per il grafico dell'andamento statistico si è utilizzata la libreria *Flot*, che mette a disposizione funzionalità interattive e i tipi di grafici più diffusi, tra i quali si è scelto quello semplice a linee con visibilità dei punti.

## **PROBLEMATICHE NELLO SVILUPPO SOFTWARE**

Durante lo sviluppo del programma si sono riscontrate alcune problematiche che hanno reso difficoltoso il suo progresso. Di seguito si espongono le maggiori difficoltà riscontrate:

1. Alcune funzioni di *Android* di default hanno bloccato le chiamate effettuate senza la codifica, pertanto si è resa necessaria la conversione in BASE 64<sup>3</sup> del nome utente e password di accesso al server web, questa modifica ha consentito al device mobile *Android* di accedere alla funzionalità dei metodi Rest.
2. Alcune chiamate AJAX da asincrone si sono dovute usare in maniera sincrona per evitare sia l'overlap delle chiamate al server sia per un'esigenza di funzionalità logica, come ad esempio nel controllo del nickname.
3. Nel caricamento delle librerie API di *Google Maps*, quando si è passato allo stato online delle applicazioni, la chiamata asincrona del caricamento della libreria ha portato a più chiamate, rendendo in tal modo instabile l'applicazione;

---

<sup>3</sup> La libreria utilizzata per la Base64 è stata ricavata da <http://www.webtoolkit.info/>

il problema presentato è stato risolto con l'utilizzo della classe *Promise Java*, per garantire l'esecuzione di una sola chiamata.

4. Il framework *Apache Cordova* ha presentato numerose carenze di debug, per compensarle durante la fase di debug dell'applicazione, si è fatto un largo utilizzo della creazione di log in console, e la visualizzazione della stessa mediante l'interfaccia del tool di sviluppo Monitor delle SDK di *Android*.

## 1. MODALITA' DI PRIMO ACCESSO

### 1.1 INTRODUZIONE

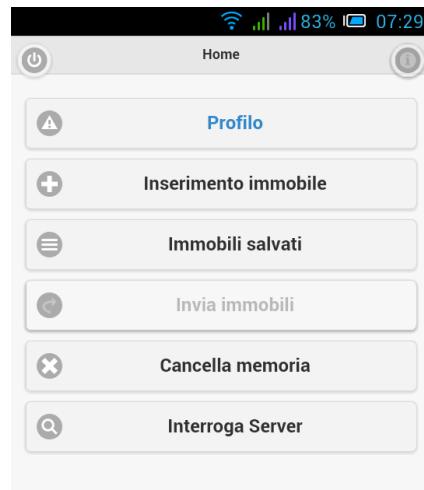
L'applicativo *Open Realtor* al momento è scaricabile solo dalla seguente url <http://realtor.dntppe.net/Realtor-debug.apk>, ma è intenzione renderla disponibile anche su App Store.

Installato il software sul proprio mobile, nel menù applicazioni del device mobile compare l'icona dell'applicazione.



**Figura 1.1** Icona di *User Realtor*

Caricando il programma, la prima schermata che compare è la seguente:

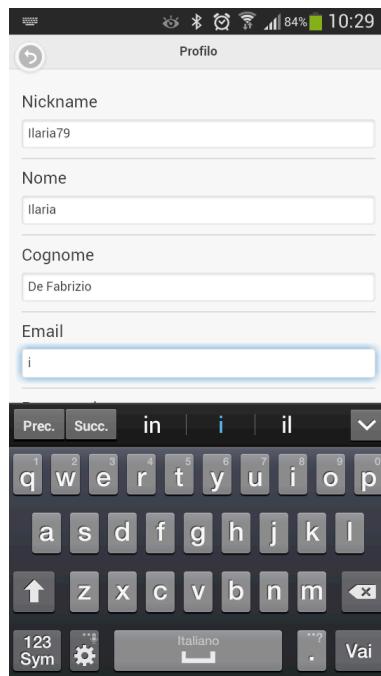


**Figura 1.2** Home page di *User Realtor* al primo accesso

Già dal primo accesso, l'utente può iniziare ad eseguire una serie di funzioni quali l'osservazione degli immobili registrati nel database; la creazione del proprio profilo; e il salvataggio dei dati di un immobile inserito; ma non si possono richiedere informazioni aggiuntive su immobili di interesse o non si possono inviare immobili inseriti, finché non si termina la procedura di creazione del proprio profilo, infatti fino a quel momento le icone corrispondenti a tali funzioni sono disabilitate.

## 1.2 CREAZIONE PROFILO UTENTE

Cliccando sul primo bottone in alto “Profilo”, l'utente può creare il proprio account per accedere alle funzioni avanzate del portale. Nella pagina del “Profilo”, i campi devono essere compilati con i propri dati, dei quali nessuno sarà visibile ad altri utenti a garanzia della privacy.



**Figura 1.3** Scheda creazione profilo

Qualora non si dovessero compilare correttamente alcuni spazi, o per dimenticanza rimangono non compilati, il programma avvisa con un messaggio di “Alert” sul particolare problema riscontrato, per cui è possibile inserire nuovamente i dati nella modalità corretta.



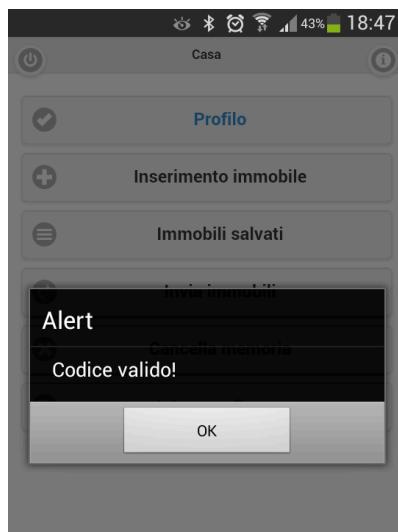
**Figura 1.4** Messaggio di errore

Nell'inserimento dei dati, per garantire l'unicità del nickname scelto, il programma controlla che il nickname non esisti già nel database, se ciò dovesse avvenire l'utente viene avvisato con un messaggio di Alert.

Completati tutti i campi, che devono contenere almeno 3 caratteri, cliccando sul bottone “Invia codice” viene inviata un'e-mail all'indirizzo specificato dall'utente nella compilazione dei dati personali, con un codice di convalida da inserire nell'apposito spazio sottostante, ciò significa che è necessario comunicare un indirizzo e-mail valido.

Questo procedimento è necessario per verificare l'identità del richiedente ed evitare che altri possano registrarsi utilizzando dati non propri. Solitamente l'e-mail viene ricevuta immediatamente, ma particolari condizioni della rete internet possono portare ad un ritardo nella consegna. Alcuni provider considerano erroneamente e-mail di questo tipo come posta indesiderata, pertanto, si consiglia di controllare anche la cartella "Posta indesiderata".

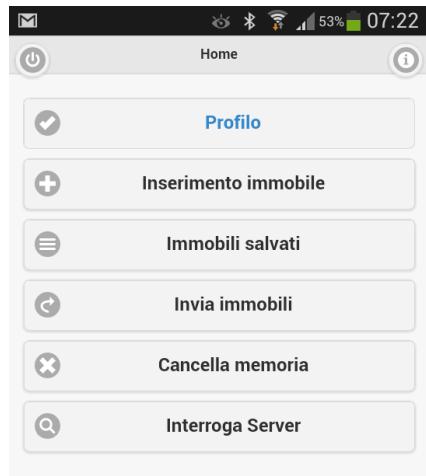
Il codice è composto da 9 caratteri alfanumerici, generati automaticamente e in modo casuale dal programma, da trascrivere così come mostrato nell'e-mail, senza spazi o modifiche tra maiuscolo o minuscolo per non generare un errore del sistema (case sensitive). Cliccando sul bottone "Convalida", se il codice è stato inserito correttamente, compare la seguente schermata:



**Figura 1.5** Messaggio di convalida codice

Con questa finestra termina il procedimento di inserimento dati e password di accesso al programma per un nuovo utente, però qualora l'utente ritenesse necessario rettificare la propria password è possibile cambiarla in qualsiasi momento, sia al primo ac-

cesso che in quelli successivi. Inoltre, da questo momento, l'utente è abilitato alla funzione per la richiesta di controllo per informazioni aggiuntive su immobili presenti nel database e alla possibilità di invio delle proprie schede immobiliari al server.



**Figura 1.6** Abilitazione bottone “i” e “Invia Immobili”

## **2. INSERIMENTO DI UN IMMOBILE**

### **2.1 GENERALITA'**

Il software permette l'inserimento dei dati di un immobile consentendo la visibilità ad altri utenti delle sue caratteristiche e far crescere così anche la popolazione statistica. Il numero di immobili che si decide di immettere sotto lo stesso nome utente non ha un limite, pertanto i professionisti possono inserire i dati dei loro clienti o sotto il proprio nome o con quello della società a cui appartengono.

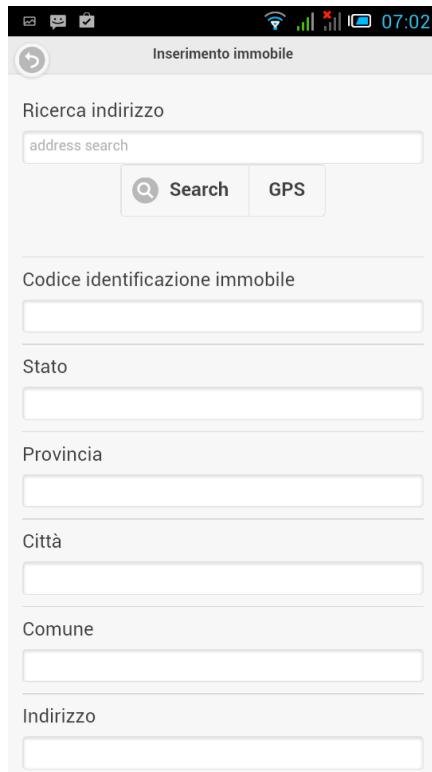
### **2.2 COME INSERIRE UN IMMOBILE**

Inserire un immobile nel programma significa innanzitutto localizzarlo geograficamente, poi referenziarlo con un codice arbitrario scelto dall'utente per distinguerlo sia dalla presenza nelle vicinanze di un altro immobile sia per catalogarlo in modo ordinato nel database. In seguito, individuata la destinazione d'uso della proprietà, si passa alla compilazione dei dati tecnici dell'immobile in oggetto: prima si individuano i dati quantitativi e poi quelli qualitativi. Al termine della compilazione dei dati tecnici, l'utente riporta il prezzo dell'immobile e, se vuole, aggiunge un numero di foto per lui necessarie, con un limite di dieci per ogni scheda.

#### **2.2.1 Ricerca indirizzo**

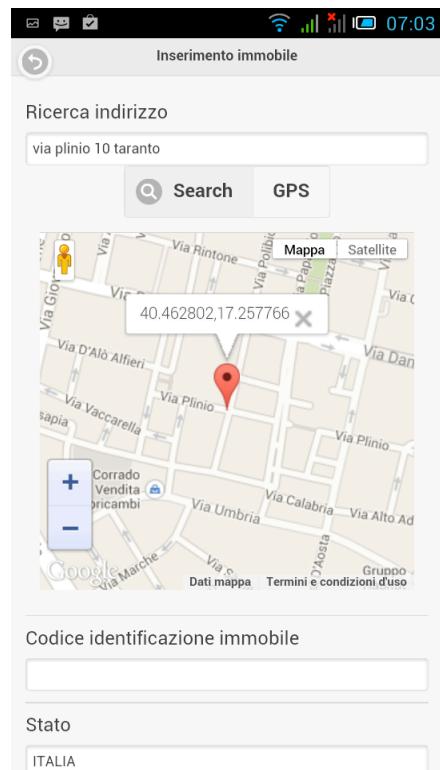
La funzione “Ricerca indirizzo”, che compare nella scheda “Inserimento immobile”, consente di cercare un indirizzo specifico o un’area per georeferenziare l’immobile.

Questa informazione è fondamentale e indispensabile prima di procedere all’invio della scheda al server.



**Figura 2.1** Scheda per la localizzazione geografica dell’immobile

La ricerca si effettua o scrivendo l’indirizzo dell’immobile interessato oppure utilizzando la posizione attuale dal GPS, poiché il programma è stato realizzato in modo che siano visualizzate le mappe di *Google*. La posizione dell’immobile è visualizzata su mappa in maniera univoca e chiara grazie alla presenza di un segnaposto, che riporta l’informazione di latitudine e longitudine, e può essere aggiornato trascinando manualmente l’indicatore, aiutandosi anche con la funzione di zoom.



**Figura 2.2** Ricerca immobile su *Google Maps*

Il programma compila automaticamente gli spazi sottostanti con i riferimenti geografici del luogo scelto, come ad esempio lo Stato, la provincia, la città, e così via.

Stato	ITALIA
Provincia	TARANTO
Città	Taranto
Comune	Taranto
Indirizzo	Via Calabria
Numero	28-44
Latitudine	40.4618633
Longitudine	17.257989599999974

**Figura 2.3** Scheda dell’immobile georeferenziato

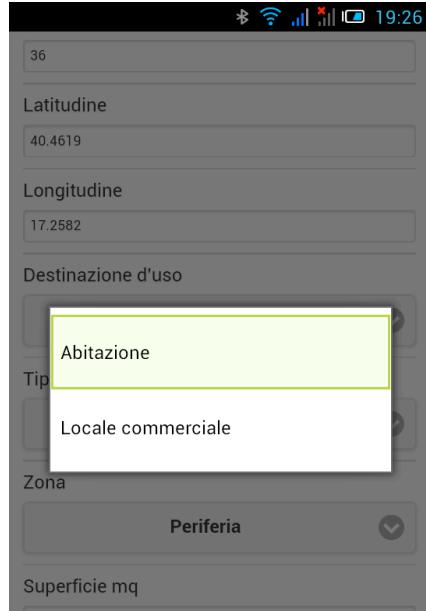
Individuata la posizione geografica, l’utente deve inserire un codice per l’identificazione dell’immobile, che può essere scelto in modo arbitrario tra caratteri alfanumerici e maiuscole. Come già detto prima, individuare un codice serve agli utenti per poter schedare in modo ordinato i propri immobili, distinguendoli da quelli individuati da altri utenti.

### 2.2.2 Inserimento caratteristiche dell’immobile

Dopo aver localizzato geograficamente l’immobile, sempre nella scheda “Inserimento immobile”, il programma procede con la richiesta dei seguenti dati tecnici:

- “Destinazione d’uso”: in base all’uso dell’immobile in oggetto, è possibile scegliere tra le opzioni di “Abitazione” o “Locale commerciale” cliccando

semplicemente sul selettore, all'apertura compare una tendina che mostra le possibilità di scelta.



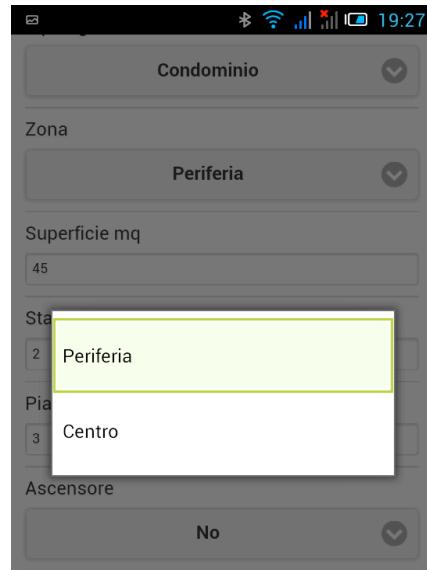
**Figura 2.4** Selettore “Destinazione d’uso”

- “Tipologia abitativa”: tra le possibili tipologie abitative si è preferito semplificare la scelta racchiudendo i casi in due preferenze “Condominio” o “Villa”, selezionabili dopo l’apertura della tendina.



**Figura 2.5** Selettore “Tipologia costruttiva”

- “Zona”: l’utente deve definire la posizione dell’immobile da inserire rispetto alla località scelta, cliccando si apre una tendina in cui è possibile selezionare “Centro” o “Periferia”.



**Figura 2.6** Selettore “Zona”

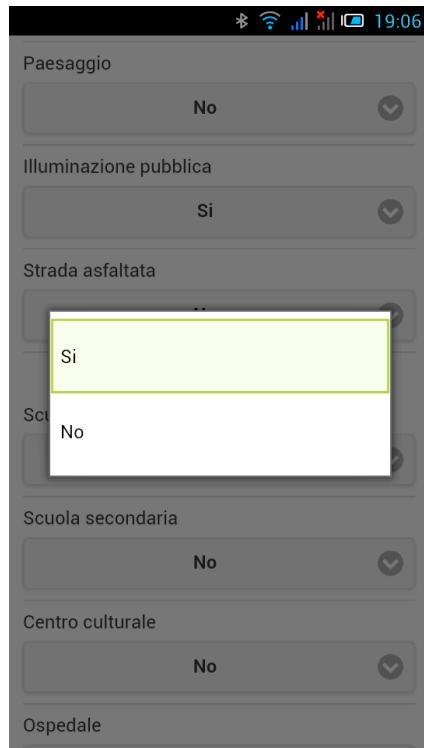
- Informazioni quantitative: il programma lascia spazio all'utente la possibilità di inserire una serie di dati tecnici che hanno un valore ben definito come “Superficie”, “Stanze”, “Piano”, e così via.

The screenshot shows a mobile application interface with the following fields:

- Superficie mq: 0
- Stanze: 0
- Piano: 0
- Ascensore: Si (selected)
- Superficie cantina mq: 0
- Superficie parcheggio mq: 0
- Superficie box mq: 0
- Superficie balconi interni mq: 0

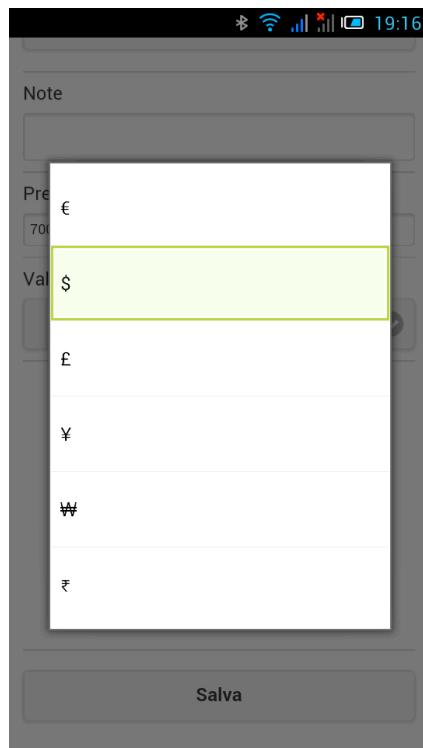
**Figura 2.7** Campi di inserimento dati quantitativi

- Informazioni qualitative: il programma lascia spazio all'utente la possibilità di inserire una serie di dati tecnici che non hanno un valore ben definito, ma sono facilmente determinabili con un'osservazione dei luoghi, esempi di tali dati sono la presenza o assenza di impianti a servizio dell'immobile, condizione dello stato manutentivo dell'immobile e dell'edificio in cui è collocato, presenza di servizi e così via.



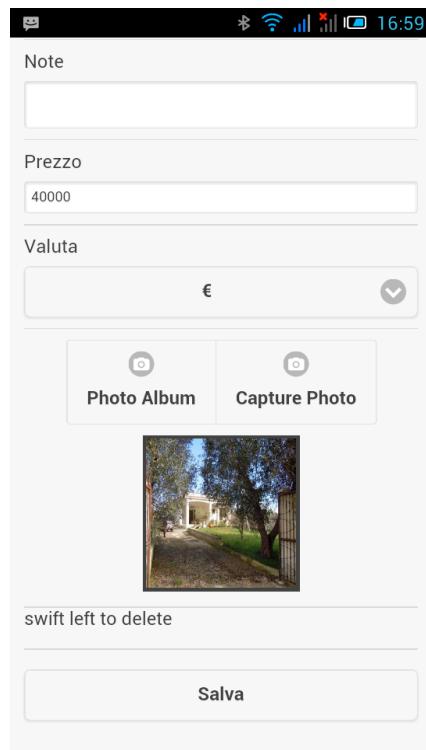
**Figura 2.8** Campi di inserimento dati qualitativi

- “Prezzo” e “Valuta”: l’ultimo spazio da compilare è il prezzo di stima dell’immobile in oggetto, tale valore può essere scelto con quale valuta è stato stimato, infatti cliccando sull’apposito bottone compare una tendina che permette di scegliere la valuta da adottare.



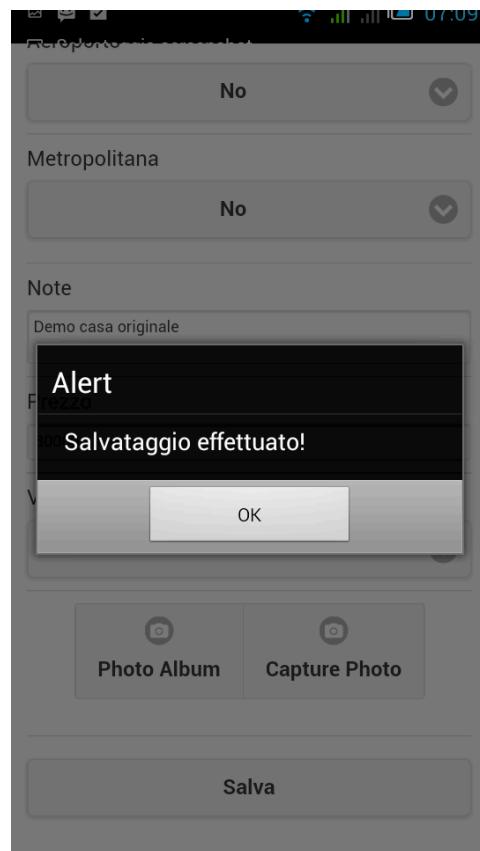
**Figura 2.9** Opzioni di scelta delle valute

- Inserimento foto: il programma consente all'utente di inserire le foto dell'immobile in numero massimo pari a 10. Le modalità di inserimento si distinguono in “Photo Album” e “Capture Photo”, nel primo caso si utilizza un archivio di foto che l'utente ha nella memoria del dispositivo utilizzato, nel secondo caso è possibile catturare istantaneamente delle foto, cancellarle qualora non fossero gradite, e se accettate inserirle nell'archivio dati dell'immobile. Qualora l'utente voglia cancellare alcune foto, e inserirne di nuove in un secondo momento, è sufficiente trascinare la foto verso sinistra.



**Figura 2.10** Inserimento immagini

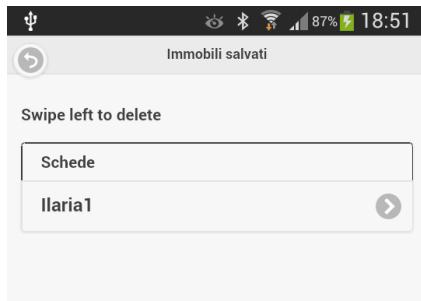
Compilati i dati e inserite le foto la funzione “Salva”, consente di immagazzinare i dati in archivio e potranno essere richiamati in qualsiasi momento. A controllo del corretto riempimento dei dati il programma fa comparire la seguente schermata:



**Figura 2.11** Messaggio di salvataggio scheda immobile nel database

### 2.3 COME RICHIAMARE UN IMMOBILE SALVATO

Nel precedente paragrafo è stata spiegata la procedura di inserimento di un nuovo immobile, per poter rivedere la scheda salvata, prima di inviarla al server e renderla visibile a tutti gli utenti, si deve interrogare la scheda “Immobili salvati”.



**Figura 2.12** Scheda per il caricamento di immobili registrati

Se si volesse cancellare la scheda è necessario trascinare con il dito verso sinistra il bottone contenete il codice immobile, invece se si scegliesse di rivedere i dati salvati è sufficiente cliccare sul bottone della scheda immobile che si vuole rivedere. Qualora si vadano a modificare alcuni dati tecnici, il programma consente il salvataggio della scheda dell'immobile sotto lo stesso codice identificativo, se invece si desidera cambiare il codice identificativo nel salvataggio, il programma genera una nuova scheda perché tale modifica viene riconosciuta come un nuovo immobile.

Si ricorda che qualora non sia disponibile una connessione ad internet, i dati salvati nella memoria del proprio dispositivo possono essere trasferiti al server in un secondo momento.

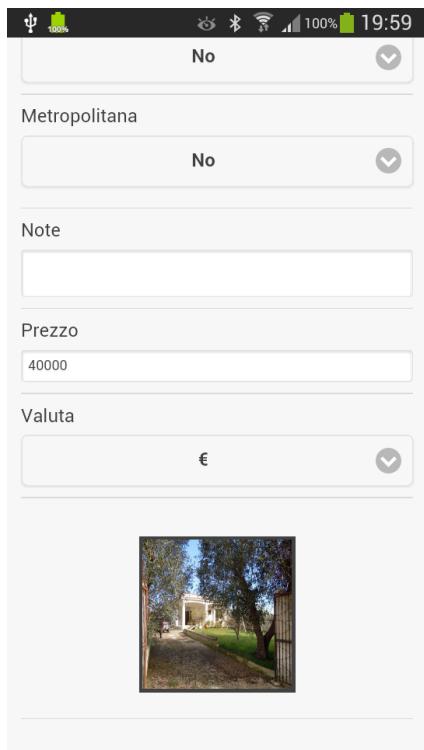
## 2.4 SALVATAGGIO IMMOBILE NEL DATABASE

Se l'utente desidera rendere visibile a tutti il proprio immobile, e non lasciarlo in memoria sul proprio mobile, può selezionare la scheda “Invia immobili”. In tale procedura il programma, prima di proseguire all’archiviazione dei dati, chiede sempre la conferma dell’invio degli immobili inseriti.



**Figura 2.13** Messaggio di conferma invio schede

L’immobile da questo momento sarà visibile a tutti gli utenti, che potranno visualizzare la scheda tecnica, ma non potranno apportare modifiche ai dati o alle foto perché non compare l’icona di salvataggio, solo l’utente che ha inserito la scheda ha questa facoltà, e dopo tale operazione può salvare la scheda e re-inviare l’immobile per aggiornare il database.



**Figura 2.13** Scheda immobile non modificabile

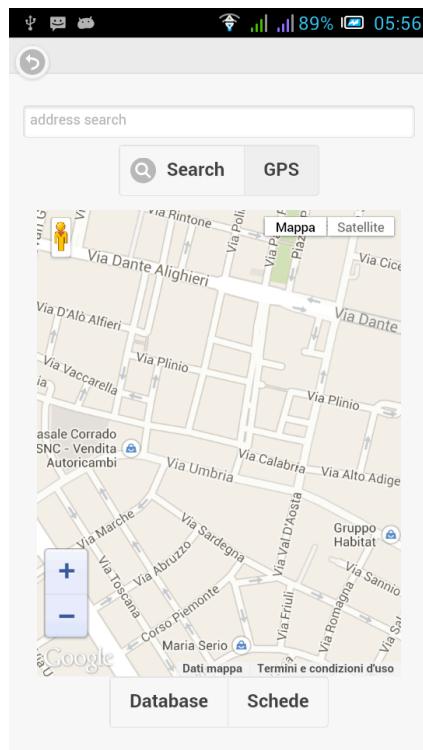
### **3. ANALISI STATISTICA DEI DATI IMMOBILIARI**

#### **3.1 GENERALITA'**

In questo capitolo si spiega come cercare i dati inseriti nel database per effettuare delle valutazioni immobiliari ed eventualmente analizzare i dati degli immobili immessi visionando tali informazioni tramite tabelle o grafici.

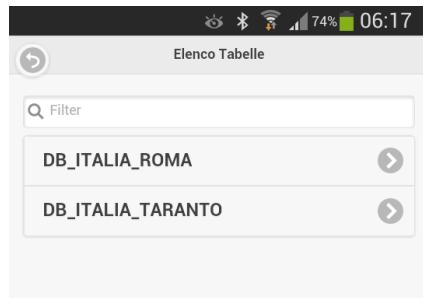
#### **3.2 FILTRO DATI**

L'utente per filtrare i dati presenti nel database può utilizzare come strumento di ricerca il quadrante delle mappe di *Google* o se preferisce consultare tutte le provincie presenti nel database con l'apposito pulsante “Database”. Nel primo caso il programma riporta nel quadrante un'area circoscritta alla via selezionata ed eventualmente si può spostare all'interno della mappa e aumentare o diminuire l'area su cui si vuole indagare, utilizzando lo strumento zoom che però è limitato ad un valore massimo per evitare un sovraccarico di richieste al server. Poi, cliccando sul bottone “Schede”, compare l'elenco degli immobili presenti nel database dell'area visualizzata.



**Figura 3.1** Inquadramento area di indagine

Nel secondo caso, decidendo di voler visionare i dati di un'area più estesa o anche tutti i dati memorizzati nel software, l'utente deve interrogare il “Database”, in tal modo comparirà l'elenco di tutte le province e aiutandosi con il campo “Filter” è possibile restringere la ricerca.



**Figura 3.3** Elenco province in memoria nel database

Con entrambi i percorsi, i due strumenti portano a visualizzare la lista delle schede, che sono filtrabili con il campo “filter”. Per visualizzare la scheda completa di dati e foto di un immobile, è sufficiente trascinare verso destra il bottone contenente la scelta effettuata.



**Figura 3.2** Visualizzazione schede presenti nel database

### 3.3 MODALITA' DI CONTATTO UTENTI

Sulla base degli immobili visionati l'utente può decidere di richiedere informazioni su una o più schede, questa operazione potrà essere effettuata contattando l'amministratore del software tramite il bottone “i” in alto.

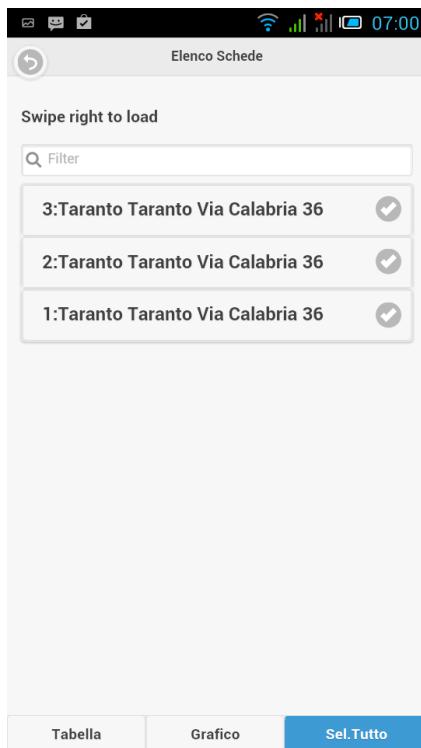
Tale procedura consente sia al proprietario dell’immobile sia all’utente interessato di rimanere nell’anonimato finché non si entra in contatto per ricevere le informazioni desiderate.



**Figura 3.4** Bottone “i” richiesta di contatto

### 3.4 ANALISI STATISTICA DEI DATI

Per portare a termine l’analisi statistica dei valori immobiliari, l’utente può decidere se selezionare solo le schede di suo interesse oppure tutte le schede che compaiono sullo schermo, digitando sul bottone “Selez. tutto”. Individuata la sua scelta chiede al programma di visualizzare i dati con tabelle o grafici sulla base delle proprie necessità.



**Figura 3.5** Caricamento schede registrate nel database

### 3.4.1 Visualizzazione tabelle

Selezionati gli immobili seguendo la procedura del precedente paragrafo, la visualizzazione statistica dei dati con il metodo tabellare avviene con il bottone “Tabella”, che fa comparire lungo le righe i singoli immobili, distinti in base alla chiave primaria univoca del database che compare nell’elenco della schermata precedente, e lungo le colonne i dati degli immobili, quali superficie, prezzo, e così via, selezionabili singolarmente, cliccando sul tasto “Columns”.

The screenshot shows a mobile application interface. At the top, there is a status bar with icons for signal strength, battery level (64%), and time (06:57). Below the status bar is a navigation bar with a back arrow and the text "Statistiche". The main area contains a table with three columns: "Nr", "surface", and "price". The table has four rows of data:

Nr	surface	price
3	45	50000
8	200	40000
5	70	80000

To the right of the table is a sidebar titled "Statistiche" with the following options:

- Nr
- surface
- price
- normalized
- price/mq
- price/sup eq

**Figura 3.6** Tabella con scelta dei dati da visualizzare

Pertanto, l’utente decide se visualizzare tali dati in contemporanea o singolarmente, così come può anche decidere di deselectarli qualora non li ritenga più utili.

### 3.4.2 Visualizzazione grafico

Qualora l’utente voglia avere una restituzione grafica sull’andamento dei prezzi degli immobili, selezionati secondo le modalità viste nel paragrafo 3.4, cliccando sul bottone “Grafico”, con i dovuti tempi di elaborazione, compare la statistica dei dati sotto forma di diagramma. La coordinata individuata sull’asse orizzontale è il tempo, definito con mese e anno, quella sull’asse verticale è il prezzo/mq oppure prezzo/sup eq (prezzo a metro quadro equivalente<sup>4</sup>), l’utente può scegliere la visualizzazione di uno dei due andamenti o di entrambi per fare un confronto anche sui due modi di rapportare il prezzo dello stesso immobile. Le due linee di interpolazione dati compaiono con colori diversi, descritti in legenda.

---

<sup>4</sup> V. nota 3 capitolo 1.



**Figura 3.6** Andamento grafici valutazioni immobiliari

Per un problema di visualizzazione grafica, considerando che i mobile non dispongono di uno schermo molto ampio, il programma non evidenzia nel grafico la coordinata di appartenenza di ciascun immobile.

#### **4. ALLEGATO: SORGENTI SVILUPPATI NELLA TESI**

Nelle pagine successive si sono allegati i sorgenti sviluppati nella tesi, di seguito si riepilogano i file con la loro funzione:

- Config.xml: file di configurazione dell'applicazione;
- Home.html: pagine in html dell'applicazione;
- Realtor.css: foglio di stile dell'applicazione;
- Realtor.js: funzioni in JavaScript utilizzate dall'applicazione;
- Language.xml: file che contiene le traduzioni nelle diverse lingue dei testi dell'applicazione;
- Index.php: file che risiede sul server web realizzato in php che fornisce l'interfaccia del server backend all'applicazione.

## config.xml

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="com.cordova.realtor" version="0.0.1"
3   xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
4     <name>Open Realtor</name>
5     <description>
6       Applicazione di statistica e stime immobiliari
7     </description>
8     <author email="dnt.ppe@gmail.com" href="http://dntppe.net">
9       Donato Pepe
10    </author>
11    <content src="home.html" />
12    <access origin="http://realtor.dntppe.net" subdomains="true" />
13
14
15    <access origin="*://*.googleapis.com/*" subdomains="true" />
16    <access origin="*://*.gstatic.com/*" subdomains="true" />
17    <access origin="*://*.google.com/*" subdomains="true" />
18    <access origin="*://*.googleusercontent.com/*" subdomains="true" />
19
20    <access origin="https://mts.googleapis.com" subdomains="true" />
21    <access origin="https://mts0.googleapis.com" subdomains="true" />
22    <access origin="https://mts1.googleapis.com" subdomains="true" />
23    <access origin="https://maps.googleapis.com" subdomains="true" />
24    <access origin="https://fonts.googleapis.com" subdomains="true" />
25    <access origin="https://maps.gstatic.com" subdomains="true" />
26    <access origin="https://csi.gstatic.com" subdomains="true" />
27    <access origin="https://themes.googleusercontent.com"
28      subdomains="true" />
29
30
31 </widget>
32
```

## home.html

```
1 <!DOCTYPE html>
2 <!--
3 Licensed to the Apache Software Foundation (ASF) under one
4 or more contributor license agreements. See the NOTICE file
5 distributed with this work for additional information
6 regarding copyright ownership. The ASF licenses this file
7 to you under the Apache License, Version 2.0 (the
8 "License"); you may not use this file except in compliance
9 with the License. You may obtain a copy of the License at
10
11 http://www.apache.org/licenses/LICENSE-2.0
12
13 Unless required by applicable law or agreed to in writing,
14 software distributed under the License is distributed on an
15 "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
16 KIND, either express or implied. See the License for the
17 specific language governing permissions and limitations
18 under the License.
19 -->
20 <html>
21   <head>
22
23     <meta charset="utf-8" />
24     <meta name="format-detection" content="telephone=no" />
25     <!-- WARNING: for iOS 7, remove the width=device-width and height=device-height
         attributes. See https://issues.apache.org/jira/browse/CB-4323 -->
26     <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1,
         minimum-scale=1, width=device-width, height=device-height" />
27     <!-- target-densitydpi=device-dpi disabilitato da precedente per incompatibilita -->
28     <title>Realtor</title>
29     <!-- Carico prima le librerie cordova -->
30     <link rel="stylesheet" type="text/css" href="jquery/jquery.mobile-1.4.2.min.css" />
31     <link rel="stylesheet" type="text/css" href="css/realtor.css" />
32     <script type="text/javascript" src="cordova.js"></script>
33
34     <script type="text/javascript" src="jquery/jquery-2.1.0.min.js"></script>
35     <script type="text/javascript" src="jquery/jquery.mobile-1.4.2.min.js"></script>
36     <script type="text/javascript" src="flot/jquery.flot.js"></script>
37     <script type="text/javascript" src="flot/jquery.flot.time.js"></script>
38     <script type="text/javascript" src="js/realtor.js"></script>
39
40   </head>
41   <body>
```

## home.html

```
42    <!--
43    <div class="app">
44    <h1>Realtor</h1>
45    <div id="deviceready" class="blink">
46
47    <p class="event listening">Connecting to Device</p>
48    <p class="event received">Device is Ready</p>
49    </div>
50    </div>
51    -->
52
53    <div data-role="page" id="homePage">
54        <div class="received" data-theme="a" data-role="header">
55            <h3 id="title">Home</h3>
56            <a id="btnExit" href="#homePage" data-rel="back" class="ui-btn ui-corner-all ui-
shadow ui-btn-a ui-icon-power ui-btn-icon-notext ui-btn-left">Close</a>
57            <a href="#" id="infoHome" class="account enableonline ui-btn ui-corner-all ui-
shadow ui-btn-a ui-icon-info ui-btn-icon-notext ui-btn-ui-btn-right">Info</a>
58
59        </div>
60        <div class="app listening" data-role="content"></div>
61        <div class="received" data-role="content">
62            <a id="btnAccount"></a>
63            <a id="btnAddImmob" data-role="button" href="#" data-icon="plus" data-
iconpos="left">Inserimento immobili</a>
64            <a id="btnListaImmob" data-role="button" href="#elencoSchede" data-icon="bars" data-
iconpos="left">Immobili salvati</a>
65            <a id="btnInviaSchede" class="account enableonline" data-role="button" href="#" data-
icon="forward" data-iconpos="left">Invia immobili</a>
66            <a id="btnClearStorage" data-role="button" href="#" data-icon="delete" data-
iconpos="left">Clear Storage</a>
67            <a id="btnSearchDB" class="enableonline" data-role="button" href="#mappafiltro" data-
icon="search" data-iconpos="left">Interroga Server</a>
68        </div>
69
70        <div data-theme="a" data-role="footer" data-position="fixed">
71            <!-- <p style="display:none;" class="event received" id="connesso">Open Realtor
is Ready</p>-->
72            <p class="event listening" id="connettendo">
73                Open Realtor connecting to device
74            </p>
75        </div>
76    </div>
```

## home.html

```
77
78     <div data-role="page" id="account">
79         <div data-theme="a" data-role="header">
80             <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
81                 back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
82             <h3 id="PageAccount">Account</h3>
83         </div>
84
85         <div data-role="content">
86             <div data-role="fieldcontain">
87                 <label for="txtnickname">Nickname</label>
88                 <input name="txtnickname" id="txtnickname" value="" type="text" value="">
89                     tabindex="1">
90                 </div>
91                 <div data-role="fieldcontain">
92                     <label for="txtfirstname" class="firstname"></label>
93                     <input name="txtfirstname" id="txtfirstname" value="" type="text" value="">
94                         tabindex="2">
95                     </div>
96                     <div data-role="fieldcontain">
97                         <label for="txtlastname" class="lastname"></label>
98                         <input name="txtlastname" id="txtlastname" value="" type="text" value="">
99                             tabindex="3">
100                         </div>
101                         <div data-role="fieldcontain">
102                             <label for="txtemailuser">Email</label>
103                             <input name="txtemailuser" id="txtemailuser" value="" type="text" value="">
104                             tabindex="4">
105                         </div>
106                         <div data-role="fieldcontain">
107                             <label for="txtpassword" class="cpassword"></label>
108                             <input name="txtpassword" id="txtpassword" value="" type="password" value="">
109                             tabindex="5">
110                         </div>
111                         <div data-role="fieldcontain">
112                             <a id="btnInviaCodice" data-role="button" href="#">Invia Codice</a>
113                         </div>
114                     <div data-role="fieldcontain">
```

## home.html

```
113      <label for="txtcode" class="code"></label>
114      <input name="txtcode" id="txtcode" value="" type="text" value=""
115          tabindex="7">
116      </div>
117      <div data-role="fieldcontain">
118          <a id="btnConvalidaCodice" data-role="button" href="#">Convalida</a>
119      </div>
120      <!--
121      <div data-theme="a" data-role="footer" data-position="fixed">
122          <h3>Realtor</h3>
123      </div>
124      -->
125  </div>
126
127  <div data-role="page" id="elencoSchede">
128      <div data-theme="a" data-role="header">
129          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
130              back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
131          <h3 id="pageImmSalvati">Immobili salvati</h3>
132      </div>
133      <div data-role="content">
134          <h3>Swipe left to delete</h3>
135          <ul id="liElencoSchede" data-role="listview" data-divider-theme="d" data-
136              inset="true"></ul>
137      </div>
138      <!--
139      <div data-theme="a" data-role="footer" data-position="fixed">
140          <h3>Realtor</h3>
141      </div>
142      -->
143  </div>
144  <div data-role="page" id="mappafiltro">
145      <div data-theme="a" data-role="header">
146          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
147              back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
148          <h3></h3>
149      </div>
150      <div data-role="content">
151          <div id="map-search2" data-role="fieldcontain" >
```

## home.html

```
152          <input name="txtIndirizzofind2" id="txtIndirizzofind2" placeholder="address  
search" value="" type="text" value="">  
153          <div data-role="controlgroup" data-type="horizontal" align="center">  
154              <a id="btnSearch2" data-role="button" data-icon="search" href="#" >Search</  
a>  
155              <a id="btnPosizione2" data-role="button" href="#" >GPS</a>  
156          </div>  
157          <div class="ui-content ui-page-theme-a" data-form="ui-page-theme-a" data-  
theme="a" role="main">  
158              <div id="map-canvas2" style="width:100%;height:400px;background-  
color:#CCC;display:none;"></div>  
159              <div data-role="controlgroup" data-type="horizontal" align="center">  
160                  <a id="btnelencoTabelle" data-role="button"  
href="#elencoTabelle">Database</a>  
161                  <a id="btnelencoSchedeServer" data-role="button"  
href="#elencoSchedeServer">Schede</a>  
162          </div>  
163      </div>  
164  </div>  
165  
166      </div>  
167  
168  </div>  
169  
170  <div data-role="page" id="elencoTabelle">  
171      <div data-theme="a" data-role="header">  
172          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-  
back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>  
173          <h3 id="pagetabelle">Elenco Tabelle</h3>  
174      </div>  
175  
176      <div data-role="content">  
177          <ul id="liElencoTable" data-role="listview" data-filter="true" data-filter-  
placeholder="Filter" data-divider-theme="d" data-inset="true"></ul>  
178      </div>  
179      <!--  
180          <div data-theme="a" data-role="footer" data-position="fixed">  
181              <h3>Realtor</h3>  
182          </div>  
183          -->  
184      </div>  
185  
186      <div data-role="page" id="elencoSchedeServer">
```

## home.html

```
187      <div data-theme="a" data-role="header">
188          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
189          <h3 id="pageelencoSchedeServer">Elenco Schede</h3>
190      </div>
191
192      <div data-role="content">
193          <h3>Swipe right to load</h3>
194          <ul id="lielencoSchedeServer" data-role="listview" data-filter="true" data-filter-
placeholder="Filter" data-divider-theme="d" data-inset="true"></ul>
195      </div>
196      <div data-theme="a" data-role="footer" data-position="fixed" >
197          <div data-role="navbar">
198              <ul>
199                  <li>
200                      <a id="btnStat" data-role="button" href="#">Tabella</a>
201                  </li>
202                  <li>
203                      <a id="btnGrafico" data-role="button" href="#">Grafico</a>
204                  </li>
205                  <li>
206                      <a id="btnAll" data-role="button" href="#">Sel.Tutto</a>
207                  </li>
208              </ul>
209          </div><!-- /navbar style="overflow:hidden;"-->
210      </div>
211  </div>
212
213  <div data-role="page" id="statistiche">
214      <div data-theme="a" data-role="header">
215          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
216          <h3 id="pagestatistiche">Statistiche</h3>
217      </div>
218
219      <div data-role="content">
220          <table data-role="table" id="table-column-toggle" data-mode="columntoggle"
class="ui-responsive table-stroke">
221              <thead>
222                  <tr id="colonnetabella">
223                      <!--
224                      <th data-priority="1">Rank</th>
225                      <th data-priority="persist">Movie Title</th>
```

## home.html

```
226      <th data-priority="2">Year</th>
227      <th data-priority="3"><abbr title="Rotten Tomato Rating">Rating</abbr></
228      th>
229      <th data-priority="4">Reviews</th>
230      -->
231      </tr>
232      </thead>
233      <tbody id="righetabella">
234      <!--
235      <tr>
236      <th>1</th>
237      <td><a href="http://en.wikipedia.org/wiki/Citizen_Kane" data-
238      rel="external">Citizen Kane</a></td>
239      <td>1941</td>
240      <td>100%</td>
241      <td>74</td>
242      </tr>
243      -->
244      </tbody>
245      </table>
246      </div>
247      <div data-theme="a" data-role="footer" data-position="fixed" ></div>
248      </div>
249
250      <div data-theme="a" data-role="header">
251          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
252          back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
253          <h3>Grafico</h3>
254          </div>
255          <div id="content">
256
257          <!--width:600px; height:300px -->
258          <div id="placeholder" class="demo-placeholder" style="float:left; ">
259
260          </div>
261
262          <p id="choices" style="float:right; margin:10px;">
263              <label>
264                  <input type="checkbox" checked="checked" name="price_mq"
id="idprice_mq" />
```

## home.html

```
265      price/mq</label>
266      <label>
267          <input type="checkbox" checked="checked" name="price_norm"
268              id="idprice_norm" />
269          price/sup eq</label>
270      </p>
271      </div>
272      <div id="footer"></div>
273  </div>
274
275  <div data-role="page" id="scheda">
276      <div data-theme="a" data-role="header">
277          <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
278              back ui-btn-icon-notext ui-btn-ui-btn-left">Back</a>
279          <a href="#" id="infoID" style="display:none;" class="account enableonline info ui-
280              btn ui-corner-all ui-shadow ui-btn-a ui-icon-info ui-btn-icon-notext ui-btn-ui-btn-right">Info</
281              a>
282          <h3 id="pageInsImmob">Inserimento immobile</h3>
283      </div>
284
285  <div data-role="content">
286      <div id="map-search" data-role="fieldcontain" style="display:none;">
287          <label class="txtserachindirizzo"></label>
288          <input name="txtIndirizzofind" id="txtIndirizzofind" placeholder="address
289              search" value="" type="text" >
290          <div data-role="controlgroup" data-type="horizontal" align="center">
291              <a id="btnSearch" data-role="button" data-icon="search" href="#" >Search</
292                  a>
293              <a id="btnPosizione" data-role="button" href="#" >GPS</a>
294          </div>
295          <div class="ui-content ui-page-theme-a" data-form="ui-page-theme-a" data-
296              theme="a" role="main">
297              <div id="map-canvas" style="width:100%;height:300px;background-
298                  color:#CCC;display:none;"></div>
299          </div>
300          <div id="geolocation"></div>
301      </div>
302      <div id=recensione></div>
303
304  <div data-role="fieldcontain">
305      <label for="codename" id="txtcodename"></label>
306      <input class="datischeda" name="codename" id="codename" type="text"
307          value="" style="text-transform:uppercase;" value="" >
```

## home.html

```
299      </div>
300      <div data-role="fieldcontain">
301          <label for="country" id="txtcountry"></label>
302          <input class="datischeda" name="country" id="country" style="text-
    transform:uppercase;" value="" >
303      </div>
304      <div data-role="fieldcontain">
305          <label for="administrative_area_level_2" id="txtadministrative_area_level_2"></
    label>
306          <input class="datischeda" name="administrative_area_level_2"
    id="administrative_area_level_2" style="text-transform:uppercase;" value="">
307      </div>
308      <div data-role="fieldcontain">
309          <label for="administrative_area_level_3" id="txtadministrative_area_level_3"></
    label>
310          <input class="datischeda" name="administrative_area_level_3"
    id="administrative_area_level_3" type="text" value="" >
311      </div>
312      <div data-role="fieldcontain">
313          <label for="locality" id="txtlocality"></label>
314          <input class="datischeda" name="locality" id="locality" type="text" value="" >
315      </div>
316      <div data-role="fieldcontain">
317          <label for="route" id="txtroute"></label>
318          <input class="datischeda" name="route" id="route" type="text" value="" >
319      </div>
320      <div data-role="fieldcontain">
321          <label for="street_number" id="txtstreet_number"></label>
322          <input class="datischeda" name="street_number" id="street_number" type="text"
    value="" >
323      </div>
324      <div data-role="fieldcontain">
325          <label for="latitude" id="txtlatitude"></label>
326          <input class="datischeda" name="latitude" id="latitude" type="number"
    value="0" >
327      </div>
328      <div data-role="fieldcontain">
329          <label for="longitude" id="txtlongitude"></label>
330          <input class="datischeda" name="longitude" id="longitude" type="number"
    value="0" >
331      </div>
332      <div data-role="fieldcontain">
333          <label for="buildingretail" class="txtbuilding"></label>
```

## home.html

```
334      <select class="datischeda" name="buildingretail" id="buildingretail"
335          type="number" value="0" >
336          <option value="0" id="txtbuildingresidential" selected></option>
337          <option value="1" id="txtbuildingretail"></option>
338      </select>
339      </div>
340      <div data-role="fieldcontain">
341          <label for="townhouse" class="txthouse"></label>
342          <select class="datischeda" name="townhouse" id="townhouse" type="number"
343              value="0" >
344              <option value="0" id="txtapartmentbuilding" selected></option>
345              <option value="1" id="txttownhouse"></option>
346          </select>
347          </div>
348          <div data-role="fieldcontain">
349              <label for="citycenter" class="txtarea"></label>
350              <select class="datischeda" name="citycenter" id="citycenter" type="number"
351                  value="0" >
352                  <option value="0" id="txtsuburb" selected></option>
353                  <option value="1" id="txtcitycenter"></option>
354          </select>
355          </div>
356          <div data-role="fieldcontain">
357              <label for="surface" id="txtsurface"></label>
358              <input class="datischeda" name="surface" id="surface" type="number"
359                  value="0" >
360          </div>
361          <div data-role="fieldcontain">
362              <label for="rooms" id="txtrooms"></label>
363              <input class="datischeda" name="rooms" id="rooms" type="number" value="0"
364          >
365          </div>
366          <div data-role="fieldcontain">
367              <label for="level" id="txtlevel"></label>
368              <input class="datischeda" name="level" id="level" type="number" value="0" >
369          </div>
370          <div data-role="fieldcontain">
371              <label for="elevator" class="txtelelevator"></label>
372              <select class="datischeda" name="elevator" id="elevator" type="number"
373                  value="0" >
374                  <option value="0" class="txtyes" selected></option>
375                  <option value="-1" class="txtno"></option>
376          </select>
```

## home.html

```
371      </div>
372      <div data-role="fieldcontain">
373          <label for="basement" id="txtbasement"></label>
374          <input class="datischeda" name="basement" id="basement" type="number"
375              value="0" >
376      </div>
377      <div data-role="fieldcontain">
378          <label for="carpark" id="txtcarpark"></label>
379          <input class="datischeda" name="carpark" id="carpark" type="number"
380              value="0" >
381      </div>
382      <div data-role="fieldcontain">
383          <label for="box" id="txtbox"></label>
384          <input class="datischeda" name="box" id="box" type="number" value="0" >
385      </div>
386      <div data-role="fieldcontain">
387          <label for="surfaceinteriorbalconies" id="txtsurfaceinteriorbalconies"></label>
388          <input class="datischeda" name="surfaceinteriorbalconies"
389              id="surfaceinteriorbalconies" type="number" value="0" >
390      </div>
391      <div data-role="fieldcontain">
392          <label for="surfaceexteriorbalconies" id="txtsurfaceexteriorbalconies"></label>
393          <input class="datischeda" name="surfaceexteriorbalconies"
394              id="surfaceexteriorbalconies" type="number" value="0" >
395      </div>
396      <div data-role="fieldcontain">
397          <label for="gardenapartment" id="txtgardenapartment"></label>
398          <input class="datischeda" name="gardenapartment" id="gardenapartment"
399              type="number" value="0" >
400          <option value="0" class="txtno" selected></option>
401          <option value="1" class="txtyes"></option>
402      </select>
403      </div>
404      <div data-role="fieldcontain">
405          <label for="waterandsanitarysystem" class="txtwaterandsanitarysystem"></label>
406          <input class="datischeda" name="waterandsanitarysystem"
407              id="waterandsanitarysystem" type="number" value="0" >
408          <option value="0" class="txtyes" selected></option>
```

home.html

```
407      <option value="-1" class="txtno"></option>
408      </select>
409  </div>
410  <div data-role="fieldcontain">
411      <select class="datischeda" name="sewerage" id="sewerage" type="number"
412      value="0" >
413          <option value="0" id="txtsewerage" selected></option>
414          <option value="-1" id="txcesspool"></option>
415          <option value="-2" id="txtnosewerage"></option>
416      </select>
417  </div>
418  <div data-role="fieldcontain">
419      <label for="numberwell" id="txtnumberwell"></label>
420      <input class="datischeda" name="numberwell" id="numberwell" type="number"
421      value="0" >
422  </div>
423  <div data-role="fieldcontain">
424      <label for="electricalsystem" class="txtelectricalsystem"></label>
425      <select class="datischeda" name="electricalsystem" id="electricalsystem"
426      type="number" value="0" >
427          <option value="0" class="txtyes" selected></option>
428          <option value="-1" class="txtno"></option>
429      </select>
430  </div>
431  <div data-role="fieldcontain">
432      <label for="gasline" class="txtgasline"></label>
433      <select class="datischeda" name="gasline" id="gasline" type="number"
434      value="0" >
435          <option value="0" class="txtyes" selected></option>
436          <option value="-1" class="txtno"></option>
437      </select>
438  </div>
439  <div data-role="fieldcontain">
440      <label for="intercomsystem" class="txtintercomsystem"></label>
441      <select class="datischeda" name="intercomsystem" id="intercomsystem"
442      type="number" value="0" >
443          <option value="0" class="txtyes" selected></option>
444          <option value="-1" class="txtno"></option>
445      </select>
446  </div>
447  <div data-role="fieldcontain">
448      <label for="allarmsystem" class="txtallarmsystem"></label>
449      <select class="datischeda" name="allarmsystem" id="allarmsystem"
```

## home.html

```
        type="number" value="0" >
445            <option value="0" class="txtno" selected></option>
446            <option value="1" class="txtyes" ></option>
447        </select>
448    </div>
449    <div data-role="fieldcontain">
450        <select class="datischeda" name="heatingsystem" id="heatingsystem"
451            type="number" value="0" >
452                <option value="0" id="txtindependentheatingsystem" selected></option>
453                <option value="-1" id="txtcentralheatingsystem"></option>
454                <option value="-2" id="txtnoheatingsystem"></option>
455            </select>
456        </div>
457        <div data-role="fieldcontain">
458            <label for="coolingsystem" class="txtcoolingsystem"></label>
459            <select class="datischeda" name="coolingsystem" id="coolingsystem"
460                type="number" value="0" >
461                <option value="0" class="txtno" selected ></option>
462                <option value="1" class="txtyes"></option>
463            </select>
464        </div>
465        <div data-role="fieldcontain">
466            <select class="datischeda" name="apartmentstateofrepair"
467                id="apartmentstateofrepair" type="number" value="0" >
468                <option value="0" id="txtgoodstate" selected ></option>
469                <option value="-1" id="txtreasonablestate"></option>
470                <option value="-2" id="txtbadstate"></option>
471            </select>
472        </div>
473        <div data-role="fieldcontain">
474            <select class="datischeda" name="homestateofrepair" id="homestateofrepair"
475                type="number" value="0" >
476                <option value="0" id="txtgoodstatehome" selected ></option>
477                <option value="-1" id="txtreasonablestatehome"></option>
478                <option value="-2" id="txtbadstatehome"></option>
479            </select>
480        </div>
481        <div data-role="fieldcontain">
```

home.html

```
482      </select>
483  </div>
484  <div data-role="fieldcontain">
485      <label for="landscape" class="txtlandscape"></label>
486      <select class="datischeda" name="landscape" id="landscape" type="number"
487      value="0" >
488          <option value="0" class="txtno" selected ></option>
489          <option value="1" class="txtyes"></option>
490      </select>
491  </div>
491  <div data-role="fieldcontain">
492      <label for="streetlighting" class="txtstreetlighting"></label>
493      <select class="datischeda" name="streetlighting" id="streetlighting"
494      type="number" value="0" >
495          <option value="0" class="txtyes" selected ></option>
496          <option value="-1" class="txtno"></option>
497      </select>
498  </div>
498  <div data-role="fieldcontain">
499      <label for="roads" class="txtasphaltroad"></label>
500      <select class="datischeda" name="roads" id="roads" type="number" value="0"
501      >
502          <option value="0" class="txtyes" selected ></option>
503          <option value="-1" class="txtno"></option>
504      </select>
504  </div>
505  <div data-role="fieldcontain">
506      <label id="txtservices" align="center"></label>
507      <div data-role="fieldcontain">
508          <label for="primaryschool" class="txtprimaryschool"></label>
509          <select class="datischeda" name="primaryschool" id="primaryschool"
510          type="number" value="0" >
511              <option value="0" class="txtyes" selected ></option>
512              <option value="-1" class="txtno"></option>
513          </select>
513  </div>
514  <div data-role="fieldcontain">
515      <label for="secondaryschool" class="txtsecondaryschool"></label>
516      <select class="datischeda" name="secondaryschool" id="secondaryschool"
517      type="number" value="0" >
518          <option value="0" class="txtno" selected></option>
519          <option value="1" class="txtyes" ></option>
519      </select>
```

## home.html

```
520      </div>
521      <div data-role="fieldcontain">
522          <label for="culturalcentre" class="txtculturalcentre"></label>
523          <select class="datischeda" name="culturalcentre" id="culturalcentre"
524              type="number" value="0" >
525              <option value="0" class="txtno" selected></option>
526              <option value="1" class="txtyes" ></option>
527          </select>
528      </div>
529      <div data-role="fieldcontain">
530          <label for="hospital" class="txthospital"></label>
531          <select class="datischeda" name="hospital" id="hospital" type="number"
532              value="0" >
533              <option value="0" class="txtyes" selected></option>
534              <option value="-1" class="txtno"></option>
535          </select>
536      </div>
537      <div data-role="fieldcontain">
538          <label for="bus" class="txtbus"></label>
539          <select class="datischeda" name="bus" id="bus" type="number" value="0" >
540              <option value="0" class="txtno" selected></option>
541              <option value="1" class="txtyes" ></option>
542          </select>
543      </div>
544      <div data-role="fieldcontain">
545          <label for="railway" class="txtrailway"></label>
546          <select class="datischeda" name="railway" id="railway" type="number"
547              value="0" >
548              <option value="0" class="txtno" selected ></option>
549              <option value="1" class="txtyes" ></option>
550          </select>
551      </div>
552      <div data-role="fieldcontain">
553          <label for="airport" class="txtairport"></label>
554          <select class="datischeda" name="airport" id="airport" type="number"
555              value="0" >
556              <option value="0" class="txtno" selected ></option>
557              <option value="1" class="txtyes" ></option>
558          </select>
559      </div>
560      <div data-role="fieldcontain">
561          <label for="subway" class="txtsubway"></label>
```

## home.html

```
559      <select class="datischeda" name="subway" id="subway" type="number"  
560          value="0" >  
561              <option value="0" class="txtno" selected ></option>  
562              <option value="1" class="txtyes" ></option>  
563          </select>  
564      </div>  
565  </div>  
566  
567      <div data-role="fieldcontain">  
568          <label for="note" id="txtnote"></label>  
569          <textarea class="datischeda" name="note" id="note" placeholder="" type="text"  
570          value=""></textarea>  
571      </div>  
572      <div data-role="fieldcontain">  
573          <label for="price" id="txtprice"></label>  
574          <input class="datischeda" name="price" id="price" type="number" value="0"  
575          placeholder="" >  
576      </div>  
577      <div data-role="fieldcontain">  
578          <label for="currency" id="txtcurrency"></label>  
579          <select class="datischeda" name="currency" id="currency" value="" type="text"  
580          value="€">  
581              <option selected>€</option>  
582              <option>$</option>  
583              <option>£</option>  
584              <option>¥</option>  
585              <option>₩</option>  
586              <option>₹</option>  
587          </select>  
588      </div>  
589      <div data-role="controlgroup" data-type="horizontal" align="center">  
590          <a id="btnAlbum" href="#" data-role="button" data-icon="camera" data-  
591          iconpos="top">Photo Album</a>  
592          <a id="btnFoto" href="#" data-role="button" data-icon="camera" data-  
593          iconpos="top">Capture Photo</a>  
594      </div>  
595      <div data-role="fieldcontain" id="gallery" align="center"></div>  
596          <label id="fotodel" style="display:none;">swift left to delete</label>  
597      </div>
```

## home.html

```
596      <div data-role="fieldcontain">
597          <a id="btnSalva" data-role="button" href="#">Salva</a>
598      </div>
599  </div>
600  <!--
601  <div data-theme="a" data-role="footer" data-position="fixed">
602      <h3>Realtor</h3>
603  </div>
604  -->
605 </div>
606
607 <!--
608 <div data-role="page" id="fotoPage">
609
610     <div data-theme="" data-role="header">
611         <h3>Foto</h3>
612     </div>
613
614     <div data-role="content">
615         <img id="fotoImmobile" src="" />
616     </div>
617
618     <div data-theme="a" data-role="footer" data-position="fixed">
619
620     </div>
621     </div>
622     -->
623     <div data-role="page" id="fotoPage" data-overlay-theme="b" data-theme="b" data-
corners="false">
624         <a href="#" data-rel="back" class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-
delete ui-btn-icon-notext ui-btn-right">Close</a>
625         <img id="fotoImmobile" class="popphoto">
626     </div>
627
628     <script type="text/javascript">
629         // app.initialize();
630         // per eccesso di sicurezza mi accerto che tutto il documento sia prima caricato
631         // anche se l'ordine di caricamento è rispettato
632         $(document).ready(function() {
633             app.initialize();
634         });
635     </script>
636 </body>
```

home.html

637  
638 </html>  
639

realtor.css

```
1 /*
2 * Licensed to the Apache Software Foundation (ASF) under one
3 * or more contributor license agreements. See the NOTICE file
4 * distributed with this work for additional information
5 * regarding copyright ownership. The ASF licenses this file
6 * to you under the Apache License, Version 2.0 (the
7 * "License"); you may not use this file except in compliance
8 * with the License. You may obtain a copy of the License at
9 *
10 * http://www.apache.org/licenses/LICENSE-2.0
11 *
12 * Unless required by applicable law or agreed to in writing,
13 * software distributed under the License is distributed on an
14 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15 * KIND, either express or implied. See the License for the
16 * specific language governing permissions and limitations
17 * under the License.
18 */
19 {
20     -webkit-tap-highlight-color: rgba(0,0,0,0); /* make transparent link selection, adjust last
   value opacity 0 to 1.0 */
21 }
22
23 body {
24     -webkit-touch-callout: none;          /* prevent callout to copy image, etc when tap to hold
   */
25     -webkit-text-size-adjust: none;        /* prevent webkit from resizing text to fit */
26     -webkit-user-select: none;           /* prevent copy paste, to allow, change 'none' to 'text' */
27     background-color:#E4E4E4;
28     background-image:linear-gradient(top, #A7A7A7 0%, #E4E4E4 51%);
29     background-image:-webkit-linear-gradient(top, #A7A7A7 0%, #E4E4E4 51%);
30     background-image:-ms-linear-gradient(top, #A7A7A7 0%, #E4E4E4 51%);
31     background-image:-webkit-gradient(
32         linear,
33         left top,
34         left bottom,
35         color-stop(0, #A7A7A7),
36         color-stop(0.51, #E4E4E4)
37 );
38     background-attachment:fixed;
39     font-family:'HelveticaNeue-Light', 'HelveticaNeue', Helvetica, Arial, sans-serif;
40     font-size:12px;
41     height:100%;
```

## realtor.css

```
42 margin:0px;
43 padding:0px;
44 /*text-transform:uppercase; disabilitato per non trasformare tutto il testo in maiuscolo
   presente per default*/
45 width:100%;
46 }
47
48 /* Portrait layout (default) */
49 .app {
50 background:url(../img/logo.png) no-repeat center top; /* 170px x 200px */
51 position:absolute;           /* position in the center of the screen */
52 left:50%; 
53 top:50%; 
54 height:50px;             /* text area height */
55 width:225px;              /* text area width */
56 text-align:center;
57 background-position:center center;
58 padding:180px 0px 0px 0px; /* image height is 200px (bottom 20px are overlapped with
   text) */
59 margin:-115px 0px 0px -112px; /* offset vertical: half of image height and text area height
   */
60                         /* offset horizontal: half of text area width */
61 }
62
63 /* Landscape layout (with min-width) */
64 @media screen and (min-aspect-ratio: 1/1) and (min-width:400px) {
65   .app {
66     background-position:center center;
67     padding:75px 0px 75px 170px; /* padding-top + padding-bottom + text area = image
       height */
68     margin:-90px 0px 0px -198px; /* offset vertical: half of image height */
69                         /* offset horizontal: half of image width and text area width */
70   }
71 }
72
73 h1 {
74   font-size:12px;
75   font-weight:normal;
76   margin:0px;
77   overflow:visible;
78   padding:0px;
79   text-align:center;
80 }
```

realtor.css

```
81
82
83
84 .event {
85   border-radius:4px;
86   -webkit-border-radius:4px;
87   font-size:12px;
88   margin:0px 30px;
89   padding:2px 0px;
90   text-align:center;
91 }
92
93 .listening {
94
95   display:block;
96 }
97
98 .received {
99
100  display:none;
101
102 }
103
104 @keyframes fade {
105   from { opacity: 1.0; }
106   50% { opacity: 0.4; }
107   to { opacity: 1.0; }
108 }
109
110 @-webkit-keyframes fade {
111   from { opacity: 1.0; }
112   50% { opacity: 0.4; }
113   to { opacity: 1.0; }
114 }
115
116 .blink {
117   animation:fade 3000ms infinite;
118   -webkit-animation:fade 3000ms infinite;
119 }
120
121 .controlgroup-textinput{
122   padding-top:.15em;
123   padding-bottom:.15em;
```

realtor.css

```
124 }  
125 .highlight {  
126 background: red;  
127 }  
128  
129
```

## realtor.js

```
1 /*
2 * Licensed to the Apache Software Foundation (ASF) under one
3 * or more contributor license agreements. See the NOTICE file
4 * distributed with this work for additional information
5 * regarding copyright ownership. The ASF licenses this file
6 * to you under the Apache License, Version 2.0 (the
7 * "License"); you may not use this file except in compliance
8 * with the License. You may obtain a copy of the License at
9 *
10 * http://www.apache.org/licenses/LICENSE-2.0
11 *
12 * Unless required by applicable law or agreed to in writing,
13 * software distributed under the License is distributed on an
14 * "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
15 * KIND, either express or implied. See the License for the
16 * specific language governing permissions and limitations
17 * under the License.
18 */
19
20 /**
21 *
22 * Base64 encode / decode
23 * http://www.webtoolkit.info/
24 *
25 */
26
27 var Base64 = {
28
29   // private property
30   _keyStr :
31     "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+=",
32
33   // public method for encoding
34   encode : function(input) {
35     var output = "";
36     var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
37     var i = 0;
38
39     input = Base64._utf8_encode(input);
40
41     while (i < input.length) {
42       chr1 = input.charCodeAt(i++);
```

## realtor.js

```
43     chr2 = input.charCodeAt(i++);
44     chr3 = input.charCodeAt(i++);
45
46     enc1 = chr1 >> 2;
47     enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
48     enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
49     enc4 = chr3 & 63;
50
51     if (isNaN(chr2)) {
52         enc3 = enc4 = 64;
53     } else if (isNaN(chr3)) {
54         enc4 = 64;
55     }
56
57     output = output + this._keyStr.charAt(enc1) + this._keyStr.charAt(enc2) +
      this._keyStr.charAt(enc3) + this._keyStr.charAt(enc4);
58
59 }
60
61 return output;
62 },
63
64 // public method for decoding
65 decode : function(input) {
66     var output = "";
67     var chr1, chr2, chr3;
68     var enc1, enc2, enc3, enc4;
69     var i = 0;
70
71     input = input.replace(/[^A-Za-z0-9\+=]/g, "");
72
73     while (i < input.length) {
74
75         enc1 = this._keyStr.indexOf(input.charAt(i++));
76         enc2 = this._keyStr.indexOf(input.charAt(i++));
77         enc3 = this._keyStr.indexOf(input.charAt(i++));
78         enc4 = this._keyStr.indexOf(input.charAt(i++));
79
80         chr1 = (enc1 << 2) | (enc2 >> 4);
81         chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
82         chr3 = ((enc3 & 3) << 6) | enc4;
83
84         output = output + String.fromCharCode(chr1);
```

## realtor.js

```
85      if (enc3 != 64) {
86          output = output + String.fromCharCode(chr2);
87      }
88      if (enc4 != 64) {
89          output = output + String.fromCharCode(chr3);
90      }
91  }
92 }
93
94
95 output = Base64._utf8_decode(output);
96
97 return output;
98
99 },
100
101 // private method for UTF-8 encoding
102 _utf8_encode : function(string) {
103     string = string.replace(/\r\n/g, "\n");
104     var utftext = "";
105
106     for (var n = 0; n < string.length; n++) {
107
108         var c = string.charCodeAt(n);
109
110         if (c < 128) {
111             utftext += String.fromCharCode(c);
112         } else if ((c > 127) && (c < 2048)) {
113             utftext += String.fromCharCode((c >> 6) | 192);
114             utftext += String.fromCharCode((c & 63) | 128);
115         } else {
116             utftext += String.fromCharCode((c >> 12) | 224);
117             utftext += String.fromCharCode(((c >> 6) & 63) | 128);
118             utftext += String.fromCharCode((c & 63) | 128);
119         }
120     }
121
122     return utftext;
123 },
124
125
126 // private method for UTF-8 decoding
127 _utf8_decode : function(utftext) {
```

## realtor.js

```
128     var string = "";
129     var i = 0;
130     var c = c1 = c2 = 0;
131
132     while (i < utftext.length) {
133
134         c = utftext.charCodeAt(i);
135
136         if (c < 128) {
137             string += String.fromCharCode(c);
138             i++;
139         } else if ((c > 191) && (c < 224)) {
140             c2 = utftext.charCodeAt(i + 1);
141             string += String.fromCharCode(((c & 31) << 6) | (c2 & 63));
142             i += 2;
143         } else {
144             c2 = utftext.charCodeAt(i + 1);
145             c3 = utftext.charCodeAt(i + 2);
146             string += String.fromCharCode(((c & 15) << 12) | ((c2 & 63) << 6) | (c3 & 63));
147             i += 3;
148         }
149
150     }
151
152     return string;
153 }
154 };
155
156 // The root URL for the RESTful services
157 var usr = "realtor";
158 var pwd = "dntppepetesi";
159 //var rootURL = "http://realtor:dntppepetesi@realtor.dntppe.net/realtor/api/";
160 var rootURL = "http://realtor.dntppe.net/realtor/api/";
161
162 var mail = {
163
164     data : {
165         to : "",
166         subject : "",
167         body : ""
168     },
169
170     send : function(dat, successCallback, failCallback) {
```

## realtor.js

```
171     console.log('mail to:' + dati);
172     $.ajax({
173         type : 'POST',
174         contentType : 'application/json',
175         //cache : false,
176         beforeSend : function(xhr) {
177             xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
178             pwd));
179         },
180         url : rootURL + 'mailto',
181         dataType : "json",
182         data : dati,
183         success : function(result, stato, errori) {
184             console.log("Chiamata successo:" + result + " " + stato + " " +
185             JSON.stringify(errori));
186             successCallback();
187             //(JSON.parse(result));
188         },
189         error : function(richiesta, stato, errori) {
190             console.log("Chiamata invio scheda al server fallita");
191             console.log(" richiesta:" + JSON.stringify(richiesta));
192             console.log(" stato:" + stato);
193             console.log(" errori:" + errori);
194             failCallback();
195         }
196     });
197
198 },
199 };
200
201 var account = {
202     //dati scheda
203     data : {
204         nickname : "",
205         firstname : "",
206         lastname : "",
207         emailuser : "",
208         password : "",
209         cpassword : "",
210         codeout : "",
211         codein : "",
```

## realtor.js

```
212  },
213
214 save : function() {
215   if (account.data.emailuser != "") { //chiave univoca
216
217     console.log('Salvo su storage user ' + JSON.stringify(account.data));
218     console.log(account.data.codein);
219     console.log(account.data.codeout);
220     app.storage.removeItem('user');
221     app.storage.setItem('user', JSON.stringify(account.data));
222     //salva i dati
223   }
224 },
225
226 check : function() {
227
228   if ((account.data.codein == account.data.codeout) && (account.data.codeout != "") &&
229     (account.data.codeout != undefined)) {
230     console.log('account valido');
231     $(".account").removeClass("ui-disabled");
232     $("#btnAccount").button({
233       icon : "check",
234       shadow : "true"
235     });
236     //$( "#btnAccount" ).attr( "data-icon","check" ).buttonMarkup('refresh');
237     //$( "#btnAccount" ).removeClass("ui-icon-alert").addClass("ui-icon-check");
238     $("#btnInviaSchede").removeClass("ui-disabled");
239
240     // controllo di esistenza account su server per sicurezza nel caso non sia andato a buon
241     // fine una precedente aggiornamento di tabella
242     account.checknick(account.data.nickname, account.data.emailuser, function(result) {
243       //successo
244       console.log("risultato checknick:" + result);
245       if (result == "false") {
246         account.addaccount();
247         // aggiorno tabella utenti su server
248         //navigator.notification.alert(mylabel("Account creato"), function() {});
249       } else {
250         account.data.codeout = "sdflj";
251         account.data.codein = "";
252         //disabilito account e costringo utente a rifarlo
253         //se mi trovo in questo caso significa che in una chiamata temporanea o dopo
254         //una precedente non andata a buon fine o ♦ stato accapputato il nickname quindi
```

## realtor.js

```
    ❓ da rifare tutto
253        account.save();
254        $(".account").addClass("ui-disabled");
255        $("#btnAccount").button({
256            icon : "alert",
257            shadow : "true"
258        });
259        navigator.notification.alert(mylabel("Nickname già esiste"), function() {
260        });
261    };
262    }, function() {
263    });
264    //errore
265
266 } else {
267     console.log('account non valido');
268     $(".account").addClass("ui-disabled");
269     $("#btnAccount").button({
270         icon : "alert",
271         shadow : "true"
272     });
273     //$( "#btnAccount" ).attr( "data-icon","alert" ).buttonMarkup('refresh');
274     //$( "#btnAccount" ).removeClass("ui-icon-check").addClass("ui-icon-alert");
275
276     $("#btnInviaSchede").addClass("ui-disabled");
277 }
278
279 },
280
281 checknick : function(name, email, successCallback, failCallback) {
282     console.log("controllo esistenza nick:" + name);
283     $.ajax({
284         async : false,
285         type : 'GET',
286         contentType : 'application/json',
287         beforeSend : function(xhr) {
288             xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
289             pwd));
290         },
291         url : rootURL + 'account/' + name + '/' + email,
292         //   dataType: "json",
293         //   data: name,
294         success : function(result, stato, errori) {
```

## realtor.js

```
294
295      console.log("Chiamata successo:" + result + " " + stato + " " +
296      JSON.stringify(errori));
297      console.log("nickname: " + result + " nick:" + name + " email:" + email);
298      successCallback(result);
299
300  },
301  error : function(richiesta, stato, errori) {
302
303      console.log("Chiamata invio scheda al server fallita");
304      console.log(" richiesta:" + JSON.stringify(richiesta));
305      console.log(" stato:" + stato);
306      console.log(" errori:" + errori);
307      failCallback();
308  }
309 });
310
311 },
312
313 load : function() {
314     //metodo per recupero dell'account
315
316     var value = app.storage.getItem('user');
317     console.log('Carico user ' + value);
318     if (value != null) {
319         console.log('Carico dati account ' + value);
320         account.data = JSON.parse(value);
321         $("#txtnickname").val(account.data.nickname);
322         $("#txtfirstname").val(account.data.firstname);
323         $("#txtlastname").val(account.data.lastname);
324         $("#txtemailuser").val(account.data.emailuser);
325
326     };
327     account.check();
328
329 },
330
331 checkEmail : function(email) {
332
333     var filter = /^[a-zA-Z0-9_\.\-]+@[([a-zA-Z0-9\-\-])+\.)+([a-zA-Z0-9]{2,4})+$/;
334
335     if (!filter.test(email) || (email == "") || (email == "undefined")) {
```

## realtor.js

```
336     console.log('non valida:' + email);
337     return false;
338 } else {
339     console.log('valida:' + email);
340     return true;
341 }
342
343 },
344
345 randomString : function() {
346     var chars =
347         "0123456789ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
348     var string_length = 8;
349     var randomstring = "";
350     for (var i = 0; i < string_length; i++) {
351         var rnum = Math.floor(Math.random() * chars.length);
352         randomstring += chars.substring(rnum, rnum + 1);
353     }
354     return (randomstring);
355 },
356
357 addaccount : function() {
358     if (app.isOnline()) {
359
360         var acc = JSON.stringify(account.data);
361         console.log('add account:' + acc);
362         $.ajax({
363             //async: false,
364             type : 'POST',
365             contentType : 'application/json',
366             beforeSend : function(xhr) {
367                 xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
368                     pwd));
369             },
370             url : rootURL + 'account',
371             dataType : "json",
372             data : acc,
373             success : function(result, stato, errori) {
374
375                 console.log("Chiamata successo:" + result + " " + stato + " " +
376                     JSON.stringify(errori));
377                 //successCallback(JSON.parse(result));
378             },
379         },
380     },
381
382     success : function(result, stato, errori) {
383
384         console.log("Chiamata successo:" + result + " " + stato + " " +
385             JSON.stringify(errori));
386         //successCallback(JSON.parse(result));
387     },
388
389     error : function(errori) {
390
391         console.log("Chiamata fallita:" + errori);
392         //errorCallback(errori);
393     }
394 }
```

## realtor.js

```
376     error : function(richiesta, stato, errori) {
377
378         console.log("Chiamata invio scheda al server fallita");
379         console.log(" richiesta:" + JSON.stringify(richiesta));
380         console.log(" stato:" + stato);
381         console.log(" errori:" + errori);
382         //failCallback();
383     }
384 );
385 }
386 }
387 };
388
389 var geocoder;
390 var map;
391
392 var quadrante_country;
393 var quadrante_administrative_area_level_2;
394 // sigla provincia
395 var loadAPIPromise;
396 var geo = {
397     //chiamata asincrona api maps google
398     loadScriptMaps : function(callback) {
399         if (app.isOnline) {
400             if (!loadAPIPromise) {
401                 var deferred = $.Deferred();
402                 $.ajax({
403                     url : 'http://www.google.com/jsapi/',
404                     dataType : "script",
405                     success : function() {
406                         google.load('maps', '3', {
407                             callback : function() {
408                                 geocoder = new google.maps.Geocoder();
409                                 console.log("Chiamata asincrona api google effettuata");
410                                 deferred.resolve();
411                             },
412                             other_params : 'sensor=false'
413                         });
414                     }
415                 });
416                 loadAPIPromise = deferred.promise();
417             }
418             loadAPIPromise.done(callback);

```

## realtor.js

```
419 } else {
420     console.log('Connessione Internet non disponibile');
421     navigator.notification.alert(mylabel("Connessione Internet non disponibile")),
422         function() {
423             });
424         );
425     },
426     search : function(input, success) {
427         geo.loadScriptMaps(function() {
428             console.log("chiamata geo");
429             if (geocoder) {
430                 console.log("geocoder caricato");
431                 geocoder.geocode(input, function(results, status) {
432                     if (status == google.maps.GeocoderStatus.OK) {
433
434                         var address = results[0].address_components;
435                         var longitude = results[0].geometry.location.e;
436                         var latitude = results[0].geometry.location.d;
437
438                         success(address, longitude, latitude);
439                     } else {
440                         console.log("Problema nella ricerca o indirizzo non trovato");
441                         navigator.notification.alert(mylabel("Problema nella ricerca o indirizzo non
442                         trovato"), function() {
443                             });
444                         });
445                         );
446                     });
447                 },
448
449             posizione : function(success) {
450                 //posizione corrente
451                 navigator.geolocation.getCurrentPosition(success, geo.onPositionError, {
452                     maximumAge : 10000,
453                     timeout : 10000,
454                     enableHighAccuracy : false
455                 });
456             },
457
458             onPositionError : function(error) {
459
```

## realtor.js

```
460     var messaggio = "";
461
462     switch (error.code) {
463
464         case PositionError.PERMISSION_DENIED:
465             messaggio = mylabel("posizione non autorizzata");
466             break;
467
468         case PositionError.POSITION_UNAVAILABLE:
469             messaggio = mylabel("posizione non disponibile");
470             break;
471
472         case PositionError.TIMEOUT:
473             messaggio = mylabel("timeout posizione");
474             break;
475     }
476     console.log('errore posizione code: ' + error.code + '\n' + 'message: ' + error.message +
477     '\n');
477     navigator.notification.alert(messaggio, function() {
478     });
479 },
480 };
481
482 var scheda = {
483
484     //dati scheda
485     data : {
486         ID : 0,
487         link : 0,
488         date : "",
489         nickname : "",
490         codename : "",
491         longitude : 0.00,
492         latitude : 0.00,
493         country : "",
494         // regione administrative_area_level_1: "",,
495         administrative_area_level_2 : "", // sigla provincia
496         administrative_area_level_3 : "", // comune
497         locality : "", // frazione
498         route : "", // strada
499         street_number : "", //numero
500         buildingretail : 0,
501         townhouse : 0,
```

## realtor.js

```
502     citycenter : 0,  
503     surface : 0,  
504     rooms : 0,  
505     level : 0,  
506     elevator : 0,  
507     basement : 0,  
508     carpark : 0,  
509     box : 0,  
510     surfaceinteriorbalconies : 0,  
511     surfaceexteriorbalconies : 0,  
512     gardenapartment : 0,  
513     sharedgarden : 0,  
514     waterandsanitarysystem : 0,  
515     watertank : 0,  
516     sewerage : 0,  
517     numberwell : 0,  
518     electricalsystem : 0,  
519     gasline : 0,  
520     intercomsystem : 0,  
521     allarmsystem : 0,  
522     heatingsystem : 0,  
523     coolingsystem : 0,  
524     apartmentstateofrepair : 0,  
525     homestateofrepair : 0,  
526     ancientbuilding : 0,  
527     landscape : 0,  
528     streetlighting : 0,  
529     roads : 0,  
530     primaryschool : 0,  
531     secondaryschool : 0,  
532     culturalcentre : 0,  
533     hospital : 0,  
534     bus : 0,  
535     railway : 0,  
536     airport : 0,  
537     subway : 0,  
538     note : "",  
539     price : 0.00,  
540     currency : "",  
541     normalized : 0,  
542     photo : []  
543 },  
544 }
```

## realtor.js

```
545 addphoto : function(image) {
546     console.log("foto:" + image);
547     var gallerypage = $("#gallery");
548     //id in index.html
549     var tempElement = $("<img class='photo' src='data:image/jpeg;base64," + image + ""
550     width='128px', height='128px', border='3'>");
551     tempElement.on("tap", function() {
552         //$("#fotoImmobile").attr("src", $(this).attr("src")).css({width: "100%"});
553         $("#fotoImmobile").attr("src", tempElement.attr("src")).css({
554             width : "100%"
555         });
556         $.mobile.changePage($("#fotoPage"), {
557             transition : 'pop',
558             role : "dialog"
559         });
560     });
561     //swipeleft sul'elemento
562     if ((scheda.data.ID == 0) || ((scheda.data.nickname == account.data.nickname) &&
563     (account.data.codein == account.data.codeout ) && (account.data.codeout != "") &&
564     (account.data.codeout != undefined))) {
565         $("#fotodel").show();
566         tempElement.on("swipeleft", function() {
567             navigator.notification.confirm("//chiedo conferma
568             mylabel("Confermi l'eliminazione della foto?"), function(buttonIndex) {
569                 if (buttonIndex == 1) {
570                     tempElement.remove();
571                     if ($("#img").length >= 10) {
572                         $("#btnAlbum").addClass("ui-disabled");
573                         $("#btnFoto").addClass("ui-disabled");
574                     } else {
575                         $("#btnAlbum").removeClass("ui-disabled");
576                         $("#btnFoto").removeClass("ui-disabled");
577                     }
578                 });
579             }
580             gallerypage.append(tempElement);
581             if ($("#img").length >= 10) {
582                 $("#btnAlbum").addClass("ui-disabled");
583                 $("#btnFoto").addClass("ui-disabled");
584             } else {
```

## realtor.js

```
585      $("#btnAlbum").removeClass("ui-disabled");
586      $("#btnFoto").removeClass("ui-disabled");
587  }
588 },
589
590 remove : function(nome) {
591     //metodo per cancellare scheda da chiave
592     if (nome != "") {
593         console.log('Cancello scheda ' + nome);
594         app.storage.removeItem(nome);
595     };
596 },
597
598 load : function(nome) {
599     //metodo per recupero scheda da chiave
600     if (nome != "") {//chiave non vuota
601         console.log('Carico scheda ' + nome);
602         var value = app.storage.getItem(nome);
603         scheda.data = JSON.parse(value);
604     };
605 },
606 },
607
608 save : function() {
609
610     $(".datischeda").each(function() {
611         if (this.id) {
612             if ($.isNumeric($(this).val())) {
613                 scheda.data[this.id] = $(this).val();
614             } else {
615                 scheda.data[this.id] = $(this).val().trim();
616             }
617             //console.log(this.id+":"+$(this).val()+" "+scheda.data[this.id]);
618         }
619     });
620     scheda.data.photo = [];
621     $(".photo").each(function() {
622         scheda.data.photo.push(this.src.replace("data:image/jpeg;base64,", ""));
623     });
624     console.log('Salvo la scheda ' + scheda.data.codename);
625     if (scheda.data.codename != "") {//chiave univoca
626         //console.log('Salvo su storage scheda ' + JSON.stringify(scheda.data));
627         app.storage.removeItem(scheda.data.codename);
```

## realtor.js

```
628     //rimuovo item nel caso sono in modifica
629     app.storage.setItem(scheda.data.codename, JSON.stringify(scheda.data));
630     //salva i dati
631     $("#gallery").empty();
632     navigator.notification.alert(mylabel("Salvataggio effettuato!"), function() {
633         });
634 } else {
635     navigator.notification.alert(mylabel("Manca codice immobile"), function() {
636         });
637 }
638 },
639
640 markermap : function(infowindow, map, marker) {
641     infowindow.setContent(marker.getPosition().toUrlValue(6));
642     infowindow.open(map, marker);
643     console.log("getposition marker:" + marker.getPosition());
644     $("#longitude").val(marker.getPosition().e);
645     $("#latitude").val(marker.getPosition().d);
646     geo.search({
647         'location' : marker.getPosition()
648     }, function(address, longitude, latitude) {
649         $(address).each(function() {
650             console.log(this.types[0] + ":" + this.long_name);
651             $("#" + this.types[0]).val(this.long_name);
652         });
653     });
654 },
655
656 mostramappa : function(longitude, latitude) {
657     geo.loadScriptMaps(function() {
658
659         var latLong = new google.maps.LatLng(latitude, longitude);
660
661         var mapOptions = {
662             zoom : 16,
663             //maxZoom:18,
664             //minZoom:14,
665             center : latLong,
666             streetViewControl : true,
667             mapTypeId : google.maps.MapTypeId.ROADMAP,
668             zoomControl : true
669         };
670         /*
```

## realtor.js

```
671      $("#map-canvas").gmap(mapOptions).show();
672      $("#map-canvas").gmap('addMarker', { 'bounds':true, 'position': latLong,'content':'Mia
673      posizione', 'draggable': true});
673      */
674
675      $("#map-canvas").show();
676      //var map=new GMap(document.getElementById("map-canvas"), mapOptions);
677      map = new google.maps.Map(document.getElementById("map-canvas"),
678      mapOptions);
679      /*
680      //Coordinate quadrante
681      google.maps.event.addListener(map, 'idle', function() {
682      quadrante=map.getBounds();
683      console.log("Coord. quadrante: "+JSON.stringify(quadrante));
684      });
685      google.maps.event.addListener(map, 'bounds_changed', function() {
686      quadrante=map.getBounds();
687      console.log("Coord. quadrante: "+JSON.stringify(quadrante));
688      });
689      */
690      var marker = new google.maps.Marker({
691          position : latLong,
692          map : map,
693          //animation: google.maps.Animation.DROP,
694          title : 'Centro',
695          draggable : true
696      );
697
698      var infowindow = new google.maps.InfoWindow();
699      scheda.markermap(infowindow, map, marker);
700      google.maps.event.addListener(marker, 'click', function() {
701          scheda.markermap(infowindow, map, marker);
702      });
703      google.maps.event.addListener(marker, "dragend", function() {
704          scheda.markermap(infowindow, map, marker);
705      });
706      });
707  },
708
709  cercaindirizzo : function(input) {
710      console.log("Cerco indirizzo: " + input);
711      geo.search({
```

## realtor.js

```
712     'address' : input
713 }, function(address, longitude, latitude) {
714     $(address).each(function() {
715         console.log(this.types[0] + ":" + this.long_name);
716         $("#" + this.types[0]).val(this.long_name);
717     });
718     console.log("Coordinate: " + longitude + "," + latitude);
719     $("#" + txtlocation_lon").val(longitude);
720     $("#" + txtlocation_lat").val(latitude);
721
722     scheda.mostramappa(longitude, latitude);
723 });
724 },
725
726 onPositionSuccess : function(position) {
727 /*
728     var element = document.getElementById('geolocation');
729
730     element.innerHTML = 'Latitude: ' + position.coords.latitude + '<br />' +
731     'Longitude: ' + position.coords.longitude + '<br />' +
732     'Altitude: ' + position.coords.altitude + '<br />' +
733     'Accuracy: ' + position.coords.accuracy + '<br />' +
734     'Altitude Accuracy: ' + position.coords.altitudeAccuracy + '<br />' +
735     'Heading: ' + position.coords.heading + '<br />' +
736     'Speed: ' + position.coords.speed + '<br />' +
737     'Timestamp: ' + position.timestamp + '<br />';
738 */
739
740     $("#" + txtlocation_lon").val(position.coords.longitude);
741     $("#" + txtlocation_lat").val(position.coords.latitude);
742
743     scheda.mostramappa(position.coords.longitude, position.coords.latitude);
744
745 },
746
747 addscheda : function(scheda, successCallback, failCallback) {
748     console.log('addscheda:' + scheda);
749     $.ajax({
750         async : false,
751         type : 'POST',
752         contentType : 'application/json',
753         beforeSend : function(xhr) {
754             xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
```

## realtor.js

```
    pwd));
755      },
756      url : rootURL + 'scheda',
757      dataType : "json",
758      data : scheda,
759      success : function(result, stato, errori) {
760
761          console.log("Chiamata successo:" + result + " " + stato + " " +
762          JSON.stringify(errori));
763          successCallback();
764          //JSON.parse(result));
765      },
766      error : function(richiesta, stato, errori) {
767
768          console.log("Chiamata invio scheda al server fallita");
769          console.log(" richiesta:" + JSON.stringify(richiesta));
770          console.log(" stato:" + stato);
771          console.log(" errori:" + errori);
772          failCallback();
773      });
774  },
775 };
776
777 var lang_def = 'english';
778
779 var language = 'english';
780
781 var mylabel = function(input) {
782     //console.log('Traduco label');
783     var output = input;
784     $.ajax({
785         url : 'language.xml',
786         async : false,
787         success : function(xml) {
788             //console.log('ok file language.xml');
789             $(xml).find('translation').each(function() {
790                 var id = $(this).attr('label');
791                 var text = $(this).find(language).text();
792                 if ((input == id) && (id != "") && (id != undefined)) {
793                     if ((text == "") || (text == undefined)) {
794                         var text = $(this).find(lang_def).text();
795                     }
796                 }
797             })
798         }
799     })
800 }
```

## realtor.js

```
796         if ((text != "") && (text != undefined)) {
797             //console.log('associo a label:' + id + ' text:' + text);
798             output = text;
799         }
800
801     }
802
803     });
804 }
805 );
806
807 return output;
808 };
809
810 /*
811
812 var googleapi = 'https://maps.googleapis.com/maps/api/js?
key=AIzaSyBf8MU_6oroWFLjnXK-
YwsNM2RDX2GiEac&v=3.exp&sensor=false&callback=loadScriptMapsOK';
813 //script.src = 'https://maps.googleapis.com/maps/api/js?
sensor=false&callback=loadScriptMapsOK';
814 //script.src = 'https://maps.googleapis.com/maps/api/js?
key=&sensor=false&callback=initialize';
815
816 */
817
818 /*
819
820 if (MapsOn) {
821 console.log("Script api google ok");
822 callback();
823 } else {
824 if (app.isOnline) {
825
826 console.log("Faccio chiamata asincrona api google");
827 var script = document.createElement('script');
828 script.type = 'text/javascript';
829 script.src = googleapi;
830
831 document.body.appendChild(script);
832
833 };
834
```

## realtor.js

```
835 };
836 */
837
838 var app = {
839   quadrante : function() {
840     //      quadrante_coord=map.getBounds();
841     app.tabella.coord[0] = map.getBounds().ga.b;
842     app.tabella.coord[1] = map.getBounds().ga.d;
843     app.tabella.coord[2] = map.getBounds().ta.d;
844     app.tabella.coord[3] = map.getBounds().ta.b;
845
846     geo.search({
847       'location' : map.getCenter()
848     }, function(address, longitude, latitude) {
849       $(address).each(function() {
850         //console.log(this.types[0] + ":" + this.long_name);
851         //$("#txt"+this.types[0]).val(this.long_name);
852         if (this.types[0] == "country") {
853           quadrante_country = this.long_name;
854           console.log("country quadrante: " + JSON.stringify(quadrante_country));
855         };
856         if (this.types[0] == "administrative_area_level_2") {
857           quadrante_administrative_area_level_2 = this.long_name;
858           console.log("provincia quadrante: " +
859             JSON.stringify(quadrante_administrative_area_level_2));
860         };
861         app.tabella.nome = ("DB_" + quadrante_country + "_" +
862           quadrante_administrative_area_level_2).toUpperCase();
863         //console.log("quadrante: "+JSON.stringify(address));
864         //    quadrante_country=address.country.long_name;
865         //
866         quadrante_administrative_area_level_2=address.administrative_area_level_2.long_name; // sigla provincia
867       });
868       console.log("Coord. quadrante: " + JSON.stringify(app.tabella.coord));
869
870     },
871     mostramappa : function(longitude, latitude) {
872       geo.loadScriptMaps(function() {
873         console.log("Mappa:" + longitude + "," + latitude);
```

realtor.js

```

874     var latLong = new google.maps.LatLng(latitude, longitude);
875
876     var mapOptions = {
877         zoom : 16,
878         maxZoom : 30,
879         minZoom : 14,
880         center : latLong,
881         streetViewControl : true,
882         mapTypeId : google.maps.MapTypeId.ROADMAP,
883         zoomControl : true
884     };
885     $("#map-canvas2").show();
886     map = new google.maps.Map(document.getElementById("map-canvas2"),
887     mapOptions);
888
889     //Coordinate quadrante
890     google.maps.event.addListener(map, 'idle', app.quadrante);
891     google.maps.event.addListener(map, 'bounds_changed', app.quadrante);
892 });
893 },
894 caricoscheda : function() {
895 /*
896     var recContainer = $("#recensione");
897     recContainer.empty();
898     if (scheda.data.link>0){
899         app.tabella.nome="DB_"+scheda.data.country.toUpperCase().trim()
900         +"_"+scheda.data.administrative_area_level_2.toUpperCase().trim();
901         var rec=<label align='center' id="" +app.tabella.nome+'/'+ scheda.data.link + "" >""
902         +mylabel("Recensione per id")+scheda.data.link + "</label>";
903         var recq=$(rec);
904         recq.on("tap", app.schedadaserver);
905         recContainer.append(recq);
906     */
907     //console.log("prova:" + (scheda.data["codename"]));
908
909     $.each(scheda.data, function(key, value) {
910         if ((key.indexOf("photo") < 0) && key) {
911             console.log(key + ": " + value);
912             $(".datischeda#" + key).val(value);
913         }

```

## realtor.js

```
914     });
915
916     $("#gallery").empty();
917     for (var j = 0; j < scheda.data.photo.length; j++) {
918         scheda.addphoto(scheda.data.photo[j]);
919     };
920     $.mobile.changePage($("#scheda"), {
921         transition : "slideup",
922         changeHash : true
923     });
924 },
925 tabella : {
926     nome : "",
927     lowlimit : 0,
928     highlimit : 50,
929     coord : [],
930 },
931 schede : {
932     nome : "",
933     lowlimit : 0,
934     highlimit : 50,
935     selezionate : [],
936 },
937
938 showdb : function(lowlimit, highlimit, successCallback, failCallback) {
939     console.log('carico tabelle dal server:');
940     $.ajax({
941         async : false,
942         type : 'GET',
943         contentType : 'application/json',
944         beforeSend : function(xhr) {
945             xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
946             pwd));
947         },
948         url : rootURL + 'showdb',
949         //dataType: "json",
950         //data: scheda,
951         success : function(result, stato, errori) {
952             console.log("Chiamata successo:" + result + " " + stato + " " +
953             JSON.stringify(errori));
954             successCallback(JSON.parse(result));
955         },
956     },
957     success : function(result, stato, errori) {
958         console.log("Chiamata successo:" + result + " " + stato + " " +
959         JSON.stringify(errori));
960         successCallback(JSON.parse(result));
961     },
962     error : function(error) {
963         console.log("Chiamata fallita: " + error);
964     }
965 }
```

## realtor.js

```
955     error : function(richiesta, stato, errori) {
956
957         console.log("Chiamata invio scheda al server fallita");
958         console.log(" richiesta:" + JSON.stringify(richiesta));
959         console.log(" stato:" + stato);
960         console.log(" errori:" + errori);
961         failCallback();
962     }
963 });
964 },
965 showschede : function(input, successCallback, failCallback) {
966     console.log('carico schede dal server :' + input);
967     $.ajax({
968         async : false,
969         type : 'POST',
970         contentType : 'application/json',
971         beforeSend : function(xhr) {
972             xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
973             pwd));
974         },
975         url : rootURL + 'showschede',
976         dataType : "json",
977         data : input,
978         success : function(result, stato, errori) {
979             console.log("Chiamata successo:" + result + " " + stato + " " +
980             JSON.stringify(errori));
981             successCallback(result);
982         },
983         error : function(richiesta, stato, errori) {
984             console.log("Chiamata ricezione scheda dal server fallita");
985             console.log(" richiesta:" + JSON.stringify(richiesta));
986             console.log(" stato:" + stato);
987             console.log(" errori:" + errori);
988             failCallback();
989         }
990     });
991 },
992 getscheda : function(id, successCallback, failCallback) {
993     console.log('carico scheda dal server:' + id);
994     $.ajax({
995         async : false,
```

## realtor.js

```
996     type : 'GET',
997     contentType : 'application/json',
998     beforeSend : function(xhr) {
999         xhr.setRequestHeader("Authorization", "Basic " + Base64.encode(usr + ":" +
1000            pwd));
1000    },
1001    url : rootURL + 'scheda/' + id,
1002    //dataType: "json",
1003    //data: scheda,
1004    success : function(result, stato, errori) {
1005        //$("#risposta").html(result);
1006        console.log("Chiamata successo:" + result + " " + stato + " " + errori);
1007        successCallback(JSON.parse(result));
1008    },
1009    error : function(richiesta, stato, errori) {
1010        //$("#risposta").html("<strong>Chiamata fallita:</strong>" + stato + " " + errori);
1011        console.log("Chiamata al server fallita");
1012        console.log(richiesta);
1013        console.log(stato);
1014        console.log(errori);
1015        failCallback();
1016    }
1017 });
1018 },
1019
1020 schedadaserver : function() {
1021     app.schede.nome = $(this).attr('id');
1022     console.log("id scheda selezionata:" + app.schede.nome);
1023     app.getscheda(app.schede.nome, function(input) {//success
1024         $.each(input, function(key, value) {
1025             if (key.indexOf("photo") < 0) {
1026                 console.log(key + ": " + value);
1027                 scheda.data[key] = input[key];
1028             };
1029         });
1030
1031     scheda.data.photo = [];
1032     if (input.photo_1) {
1033         scheda.data.photo.push(input.photo_1);
1034     };
1035     if (input.photo_2) {
1036         scheda.data.photo.push(input.photo_2);
1037     };
1038 }
```

## realtor.js

```
1038     if (input.photo_3) {
1039         scheda.data.photo.push(input.photo_3);
1040     };
1041     if (input.photo_4) {
1042         scheda.data.photo.push(input.photo_4);
1043     };
1044     if (input.photo_5) {
1045         scheda.data.photo.push(input.photo_5);
1046     };
1047     if (input.photo_6) {
1048         scheda.data.photo.push(input.photo_6);
1049     };
1050     if (input.photo_7) {
1051         scheda.data.photo.push(input.photo_7);
1052     };
1053     if (input.photo_8) {
1054         scheda.data.photo.push(input.photo_8);
1055     };
1056     if (input.photo_9) {
1057         scheda.data.photo.push(input.photo_9);
1058     };
1059     if (input.photo_10) {
1060         scheda.data.photo.push(input.photo_10);
1061     };
1062
1063     app.caricoscheda();
1064 }, function() {//error
1065 });
1066 },
1067
1068 onCameraSuccess : function(image) {
1069     console.log("Dimensione foto in caratteri:" + image.length);
1070     scheda.addphoto(image);
1071 },
1072 onCameraError : function(errorMessage) {
1073
1074     navigator.notification.alert(errorMessage, function() {
1075 });
1076
1077 },
1078 online : function() {
1079     console.log('abilita invia schede');
1080     $("#btnInviaSchede").removeClass("ui-disabled");
```

## realtor.js

```
1081     $("#btnAccount").removeClass("ui-disabled");
1082     $("#btnSearchDB").removeClass("ui-disabled");
1083     $(".enableonline").removeClass("ui-disabled");
1084     //$("#map-search").show();
1085     $(".showonline").show();
1086     geo.loadScriptMaps(function() {
1087     });
1088     account.check();
1089
1090 },
1091
1092 offline : function() {
1093     console.log('disabilita invia schede');
1094     $("#btnInviaSchede").addClass("ui-disabled");
1095     $("#btnAccount").addClass("ui-disabled");
1096     $("#btnSearchDB").addClass("ui-disabled");
1097     //$("#map-search").hide();
1098     $(".enableonline").addClass("ui-disabled");
1099     $(".showonline").hide();
1100
1101 },
1102
1103 // Update DOM on a Received Event
1104 receivedEvent : function(id) {
1105     /*
1106     var parentElement = document.getElementById(id);
1107     var listeningElement = parentElement.querySelector('.listening');
1108     var receivedElement = parentElement.querySelector('.received');
1109
1110     listeningElement.setAttribute('style', 'display:none;');
1111     receivedElement.setAttribute('style', 'display:block;');
1112     */
1113
1114     $(".listening").css("display", "none");
1115     $(".received").css("display", "block");
1116
1117     console.log('Received Event: ' + id);
1118 },
1119
1120 storage : window.localStorage, //memorizzazione permanente , window.sessionStorage nel
caso per sessione
1121
1122 //stato online dell'applicazione
```

## realtor.js

```
1123  isOnline : function() {
1124      var networkState = navigator.connection.type;
1125      /*
1126          if (((networkState != Connection.NONE) && (networkState != Connection.UNKNOWN))) {
1127
1128              } else {
1129                  return false;
1130              }
1131
1132          */
1133
1134      return ((networkState != Connection.NONE) && (networkState != Connection.UNKNOWN));
1135      /*
1136          navigator.network.isReachable("www.google.com",function() {
1137              if ( reachability.remoteHostStatus == NetworkStatus.NOT_REACHABLE ) {
1138                  console.log("You appear to be offline.");
1139                  // Close app
1140              } else {
1141                  // Inizialize App
1142              }
1143              return (( reachability.remoteHostStatus != NetworkStatus.NOT_REACHABLE ));
1144          }, {isIpAddress:false});
1145      */
1146  },
1147
1148  exit : function() {
1149
1150      navigator.notification.confirm(mylabel("Vuoi uscire dall'applicazione?"),
1151          function(buttonIndex) {
1152
1153          if (buttonIndex == 1) {
1154              console.log('Uscita');
1155              navigator.app.exitApp();
1156          });
1157
1158  },
1159  // deviceready Event Handler
1160  //
1161  // The scope of 'this' is the event. In order to call the 'receivedEvent'
1162  // function, we must explicitly call 'app.receivedEvent(...);'
```

realtor.js

```
1163  onDeviceReady : function() {
1164
1165      $.mobile.allowCrossDomainPages = true;
1166      $.support.cors = true;
1167      //$.mobile.page.prototype.options.domCache = false;
1168      /*
1169      jQuery('div').live('pagehide', function(event, ui){
1170          var page = jQuery(event.target);
1171
1172          if(page.attr('data-cache') == 'never'){
1173              page.remove();
1174          };
1175      });
1176      */
1177      navigator.globalization.getPreferredLanguage(function(lang) {
1178          console.log('lang: ' + lang.value + '\n');
1179          language = lang.value;
1180      }, function() {
1181          console.log('Error getting language\n');
1182      });
1183
1184      $(function() {
1185          //console.log('Traduco id html');
1186          $.ajax({
1187              url : 'language.xml',
1188              success : function(xml) {
1189                  //console.log('ok file language.xml');
1190                  $(xml).find('translation').each(function() {
1191                      var id = $(this).attr('id');
1192                      var text = $(this).find(language).text();
1193
1194                      if ((id != "") && (id != undefined)) {
1195                          if ((text == "") || (text == "undefined")) {
1196                              var text = $(this).find(lang_def).text();
1197                          }
1198                          if ((text != "") && (text != undefined)) {
1199                              //console.log('associo a id:' + id + ' text:' + text);
1200                              $("#" + id).each(function() {
1201                                  $(this).html(text);
1202                              });
1203                              $(".#" + id).each(function() {
1204                                  $(this).html(text);
1205                              });
1206                          }
1207                      }
1208                  });
1209              }
1210          });
1211      });
1212  });
1213
1214  
```

## realtor.js

```
1206          }
1207      }
1208  });
1209  }
1210  });
1211  });
1212
1213  $("#btnClearStorage").on("tap", function() {
1214      navigator.notification.confirm("//chiedo conferma
1215      mylabel("Reset totale?"), function(buttonIndex) {
1216          if (buttonIndex == 1) {
1217              app.storage.clear();
1218          }
1219      });
1220
1221  });
1222
1223  //assegnamento della gestione dell'evento tap, l'evento corrispondente al tocco sul
1224  pulsante
1225  $("#btnInviaCodice").on("tap", function() {
1226      if (($("#txtnickname").val().trim().length >= 3) && ($
1227      ("#txtfirstname").val().trim().length >= 3) && ($("#txtlastname").val().trim().length >= 3)
1228      && ($("#txtemailuser").val().trim().length >= 3)) {
1229          if (app.isOnline()) {
1230              console.log('Connessione Internet disponibile');
1231
1232              account.data.firstname = ($("#txtfirstname").val().trim());
1233              account.data.lastname = ($("#txtlastname").val().trim());
1234              account.data.emailuser = ($("#txtemailuser").val().trim());
1235
1236              if (account.checkEmail(account.data.emailuser)) {
1237                  account.data.password = ($("#txtpassword").val().trim());
1238                  account.data.cpassword = ($("#txtcpassword").val().trim());
1239                  if (account.data.cpassword == account.data.password) {
1240
1241                      mail.data.to = account.data.emailuser;
1242                      mail.data.subject = mylabel("Codice di convalida account per l'app Realtor");
1243                      account.data.codeout = account.randomString();
1244                      mail.data.body = 'Code: ' + account.data.codeout;
1245                      account.save();
1246                      //console.log('mailt to:'+' '+mail.data.to+' '+mail.data.subject+
1247                      '+mail.data.body);
1248                      value = JSON.stringify(mail.data);
```

realtor.js

```
1245     console.log(value);
1246     mail.send(value, function() {
1247         console.log('Email inviat');
1248
1249         navigator.notification.alert(mylabel("Email inviat"), function() {
1250             });
1251         }, function() {
1252             //nel caso di errore
1253             console.log('Email not inviat');
1254             navigator.notification.alert(mylabel("Email not inviat"), function() {
1255                 });
1256             });
1257
1258     } else {
1259         navigator.notification.alert(mylabel("Password non corrispondono!"),
1260             function() {
1261                 });
1262             );
1263         } else {
1264             navigator.notification.alert(mylabel("Email non valida!"), function() {
1265                 });
1266             );
1267         } else {
1268             console.log('Connessione Internet non disponibile');
1269             navigator.notification.alert(mylabel("Connessione Internet non disponibile!"),
1270                 function() {
1271                     );
1272             } else {
1273                 navigator.notification.alert(mylabel("Tutti i campi devono essere pieni!"), function()
1274                 );
1275             );
1276         });
1277
1278 $("#btnConvalidaCodice").on("tap", function() {
1279     if (app.isOnline()) {
1280         console.log('Connessione Internet disponibile preparo invio schede');
1281         account.data.nickname = $("#txtnickname").val().trim();
1282         if ($("#txtcode").val().trim() == account.data.codeout) {
1283             navigator.notification.alert(mylabel("Codice valido"), function() {
1284                 });
1285         }
1286     }
1287
1288     if (account.data.nickname != "" & account.data.codeout != "") {
1289         var data = "nickname=" + account.data.nickname + "&codeout=" + account.data.codeout;
1290         $.post("http://www.mysite.it/invia_email.php", data, function(response) {
1291             if (response == "OK") {
1292                 navigator.notification.alert("Email inviat");
1293             } else {
1294                 navigator.notification.alert("Email non inviat");
1295             }
1296         });
1297     }
1298
1299     else {
1300         navigator.notification.alert("Devi inserire nickname e codice");
1301     }
1302
1303     });
1304
1305     });
1306
1307     });
1308
1309     });
1310
1311     });
1312
1313     });
1314
1315     });
1316
1317     });
1318
1319     });
1320
1321     });
1322
1323     });
1324
1325     });
1326
1327     });
1328
1329     });
1330
1331     });
1332
1333     });
1334
1335     });
1336
1337     });
1338
1339     });
1340
1341     });
1342
1343     });
1344
1345     });
1346
1347     });
1348
1349     });
1350
1351     });
1352
1353     });
1354
1355     });
1356
1357     });
1358
1359     });
1360
1361     });
1362
1363     });
1364
1365     });
1366
1367     });
1368
1369     });
1370
1371     });
1372
1373     });
1374
1375     });
1376
1377     });
1378
1379     });
1380
1381     });
1382
1383     });
1384
1385     });
1386
1387     });
1388
1389     });
1390
1391     });
1392
1393     });
1394
1395     });
1396
1397     });
1398
1399     });
1400
1401     });
1402
1403     });
1404
1405     });
1406
1407     });
1408
1409     });
1410
1411     });
1412
1413     });
1414
1415     });
1416
1417     });
1418
1419     });
1420
1421     });
1422
1423     });
1424
1425     });
1426
1427     });
1428
1429     });
1430
1431     });
1432
1433     });
1434
1435     });
1436
1437     });
1438
1439     });
1440
1441     });
1442
1443     });
1444
1445     });
1446
1447     });
1448
1449     });
1450
1451     });
1452
1453     });
1454
1455     });
1456
1457     });
1458
1459     });
1460
1461     });
1462
1463     });
1464
1465     });
1466
1467     });
1468
1469     });
1470
1471     });
1472
1473     });
1474
1475     });
1476
1477     });
1478
1479     });
1480
1481     });
1482
1483     });
1484
1485     });
1486
1487     });
1488
1489     });
1490
1491     });
1492
1493     });
1494
1495     });
1496
1497     });
1498
1499     });
1500
1501     });
1502
1503     });
1504
1505     });
1506
1507     });
1508
1509     });
1510
1511     });
1512
1513     });
1514
1515     });
1516
1517     });
1518
1519     });
1520
1521     });
1522
1523     });
1524
1525     });
1526
1527     });
1528
1529     });
1530
1531     });
1532
1533     });
1534
1535     });
1536
1537     });
1538
1539     });
1540
1541     });
1542
1543     });
1544
1545     });
1546
1547     });
1548
1549     });
1550
1551     });
1552
1553     });
1554
1555     });
1556
1557     });
1558
1559     });
1560
1561     });
1562
1563     });
1564
1565     });
1566
1567     });
1568
1569     });
1570
1571     });
1572
1573     });
1574
1575     });
1576
1577     });
1578
1579     });
1580
1581     });
1582
1583     });
1584
1585     });
1586
1587     });
1588
1589     });
1590
1591     });
1592
1593     });
1594
1595     });
1596
1597     });
1598
1599     });
1599
1600     });
1601
1602     });
1603
1604     });
1605
1606     });
1607
1608     });
1609
1610     });
1611
1612     });
1613
1614     });
1615
1616     });
1617
1618     });
1619
1620     });
1621
1622     });
1623
1624     });
1625
1626     });
1627
1628     });
1629
1630     });
1631
1632     });
1633
1634     });
1635
1636     });
1637
1638     });
1639
1640     });
1641
1642     });
1643
1644     });
1645
1646     });
1647
1648     });
1649
1650     });
1651
1652     });
1653
1654     });
1655
1656     });
1657
1658     });
1659
1660     });
1661
1662     });
1663
1664     });
1665
1666     });
1667
1668     });
1669
1670     });
1671
1672     });
1673
1674     });
1675
1676     });
1677
1678     });
1679
1680     });
1681
1682     });
1683
1684     });
1685
1686     });
1687
1688     });
1689
1690     });
1691
1692     });
1693
1694     });
1695
1696     });
1697
1698     });
1699
1699
1700     });
1701
1702     });
1703
1704     });
1705
1706     });
1707
1708     });
1709
1710     });
1711
1712     });
1713
1714     });
1715
1716     });
1717
1718     });
1719
1720     });
1721
1722     });
1723
1724     });
1725
1726     });
1727
1728     });
1729
1729
1730     });
1731
1732     });
1733
1734     });
1735
1736     });
1737
1738     });
1739
1739
1740     });
1741
1742     });
1743
1744     });
1745
1746     });
1747
1748     });
1749
1749
1750     });
1751
1752     });
1753
1754     });
1755
1756     });
1757
1758     });
1759
1759
1760     });
1761
1762     });
1763
1764     });
1765
1766     });
1767
1768     });
1769
1769
1770     });
1771
1772     });
1773
1774     });
1775
1776     });
1777
1778     });
1779
1779
1780     });
1781
1782     });
1783
1784     });
1785
1786     });
1787
1788     });
1789
1789
1790     });
1791
1792     });
1793
1794     });
1795
1796     });
1797
1798     });
1799
1799
1800     });
1801
1802     });
1803
1804     });
1805
1806     });
1807
1808     });
1809
1809
1810     });
1811
1812     });
1813
1814     });
1815
1816     });
1817
1818     });
1819
1819
1820     });
1821
1822     });
1823
1824     });
1825
1826     });
1827
1827
1828     });
1829
1830     });
1831
1832     });
1833
1834     });
1835
1836     });
1837
1837
1838     });
1839
1840     });
1841
1842     });
1843
1844     });
1845
1845
1846     });
1847
1848     });
1849
1850     });
1851
1851
1852     });
1853
1854     });
1855
1856     });
1857
1858     });
1859
1859
1860     });
1861
1862     });
1863
1864     });
1865
1866     });
1867
1867
1868     });
1869
1870     });
1871
1872     });
1873
1873
1874     });
1875
1876     });
1877
1878     });
1879
1879
1880     });
1881
1882     });
1883
1884     });
1885
1886     });
1887
1887
1888     });
1889
1890     });
1891
1892     });
1893
1893
1894     });
1895
1896     });
1897
1898     });
1899
1899
1900     });
1901
1902     });
1903
1904     });
1905
1906     });
1907
1907
1908     });
1909
1910     });
1911
1911
1912     });
1913
1914     });
1915
1916     });
1917
1917
1918     });
1919
1920     });
1921
1921
1922     });
1923
1924     });
1925
1925
1926     });
1927
1928     });
1929
1929
1930     });
1931
1932     });
1933
1933
1934     });
1935
1936     });
1937
1937
1938     });
1939
1940     });
1941
1941
1942     });
1943
1944     });
1945
1945
1946     });
1947
1948     });
1949
1949
1950     });
1951
1952     });
1953
1953
1954     });
1955
1956     });
1957
1957
1958     });
1959
1960     });
1961
1961
1962     });
1963
1964     });
1965
1965
1966     });
1967
1968     });
1969
1969
1970     });
1971
1972     });
1973
1973
1974     });
1975
1976     });
1977
1977
1978     });
1979
1980     });
1981
1981
1982     });
1983
1984     });
1985
1985
1986     });
1987
1988     });
1989
1989
1990     });
1991
1992     });
1993
1993
1994     });
1995
1996     });
1997
1997
1998     });
1999
2000     });
2001
2001
2002     });
2003
2004     });
2005
2005
2006     });
2007
2008     });
2009
2009
2010     });
2011
2012     });
2013
2013
2014     });
2015
2016     });
2017
2017
2018     });
2019
2020     });
2021
2021
2022     });
2023
2024     });
2025
2025
2026     });
2027
2028     });
2029
2029
2030     });
2031
2032     });
2033
2033
2034     });
2035
2036     });
2037
2037
2038     });
2039
2040     });
2041
2041
2042     });
2043
2044     });
2045
2045
2046     });
2047
2048     });
2049
2049
2050     });
2051
2052     });
2053
2053
2054     });
2055
2056     });
2057
2057
2058     });
2059
2060     });
2061
2061
2062     });
2063
2064     });
2065
2065
2066     });
2067
2068     });
2069
2069
2070     });
2071
2072     });
2073
2073
2074     });
2075
2076     });
2077
2077
2078     });
2079
2080     });
2081
2081
2082     });
2083
2084     });
2085
2085
2086     });
2087
2088     });
2089
2089
2090     });
2091
2092     });
2093
2093
2094     });
2095
2096     });
2097
2097
2098     });
2099
2100     });
2101
2101
2102     });
2103
2104     });
2105
2105
2106     });
2107
2108     });
2109
2109
2110     });
2111
2112     });
2113
2113
2114     });
2115
2116     });
2117
2117
2118     });
2119
2120     });
2121
2121
2122     });
2123
2124     });
2125
2125
2126     });
2127
2128     });
2129
2129
2130     });
2131
2132     });
2133
2133
2134     });
2135
2136     });
2137
2137
2138     });
2139
2140     });
2141
2141
2142     });
2143
2144     });
2145
2145
2146     });
2147
2148     });
2149
2149
2150     });
2151
2152     });
2153
2153
2154     });
2155
2156     });
2157
2157
2158     });
2159
2160     });
2161
2161
2162     });
2163
2164     });
2165
2165
2166     });
2167
2168     });
2169
2169
2170     });
2171
2172     });
2173
2173
2174     });
2175
2176     });
2177
2177
2178     });
2179
2180     });
2181
2181
2182     });
2183
2184     });
2185
2185
2186     });
2187
2188     });
2189
2189
2190     });
2191
2192     });
2193
2193
2194     });
2195
2196     });
2197
2197
2198     });
2199
2200     });
2201
2201
2202     });
2203
2204     });
2205
2205
2206     });
2207
2208     });
2209
2209
2210     });
2211
2212     });
2213
2213
2214     });
2215
2216     });
2217
2217
2218     });
2219
2220     });
2221
2221
2222     });
2223
2224     });
2225
2225
2226     });
2227
2228     });
2229
2229
2230     });
2231
2232     });
2233
2233
2234     });
2235
2236     });
2237
2237
2238     });
2239
2240     });
2241
2241
2242     });
2243
2244     });
2245
2245
2246     });
2247
2248     });
2249
2249
2250     });
2251
2252     });
2253
2253
2254     });
2255
2256     });
2257
2257
2258     });
2259
2260     });
2261
2261
2262     });
2263
2264     });
2265
2265
2266     });
2267
2268     });
2269
2269
2270     });
2271
2272     });
2273
2273
2274     });
2275
2276     });
2277
2277
2278     });
2279
2280     });
2281
2281
2282     });
2283
2284     });
2285
2285
2286     });
2287
2288     });
2289
2289
2290     });
2291
2292     });
2293
2293
2294     });
2295
2296     });
2297
2297
2298     });
2299
2300     });
2301
2301
2302     });
2303
2304     });
2305
2305
2306     });
2307
2308     });
2309
2309
2310     });
2311
2312     });
2313
2313
2314     });
2315
2316     });
2317
2317
2318     });
2319
2320     });
2321
2321
2322     });
2323
2324     });
2325
2325
2326     });
2327
2328     });
2329
2329
2330     });
2331
2332     });
2333
2333
2334     });
2335
2336     });
2337
2337
2338     });
2339
2340     });
2341
2341
2342     });
2343
2344     });
2345
2345
2346     });
2347
2348     });
2349
2349
2350     });
2351
2352     });
2353
2353
2354     });
2355
2356     });
2357
2357
2358     });
2359
2360     });
2361
2361
2362     });
2363
2364     });
2365
2365
2366     });
2367
2368     });
2369
2369
2370     });
2371
2372     });
2373
2373
2374     });
2375
2376     });
2377
2377
2378     });
2379
2380     });
2381
2381
2382     });
2383
2384     });
2385
2385
2386     });
2387
2388     });
2389
2389
2390     });
2391
2392     });
2393
2393
2394     });
2395
2396     });
2397
2397
2398     });
2399
2400     });
2401
2401
2402     });
2403
2404     });
2405
2405
2406     });
2407
2408     });
2409
2409
2410     });
2411
2412     });
2413
2413
2414     });
2415
2416     });
2417
2417
2418     });
2419
2420     });
2421
2421
2422     });
2423
2424     });
2425
2425
2426     });
2427
2428     });
2429
2429
2430     });
2431
2432     });
2433
2433
2434     });
2435
2436     });
2437
2437
2438     });
2439
2440     });
2441
2441
2442     });
2443
2444     });
2445
2445
2446     });
2447
2448     });
2449
2449
2450     });
2451
2452     });
2453
2453
2454     });
2455
2456     });
2457
2457
2458     });
2459
2460     });
2461
2461
2462     });
2463
2464     });
2465
2465
2466     });
2467
2468     });
2469
2469
2470     });
2471
2472     });
2473
2473
2474     });
2475
2476     });
2477
2477
2478     });
2479
2480     });
2481
2481
2482     });
2483
2484     });
2485
2485
2486     });
2487
2488     });
2489
2489
2490     });
2491
2492     });
2493
2493
2494     });
2495
2496     });
2497
2497
2498     });
2499
2500     });
2501
2501
2502     });
2503
2504     });
2505
2505
2506     });
2507
2508     });
2509
2509
2510     });
2511
2512     });
2513
2513
2514     });
2515
2516     });
2517
2517
2518     });
2519
2520     });
2521
2521
2522     });
2523
2524     });
2525
2525
2526     });
2527
2528     });
2529
2529
2530     });
2531
2532     });
2533
2533
2534     });
2535
2536     });
2537
2537
2538     });
2539
2540     });
2541
2541
2542     });
2543
2544     });
2545
2545
2546     });
2547
2548     });
2549
2549
2550     });
2551
2552     });
2553
2553
2554     });
2555
2556     });
2557
2557
2558     });
2559
2560     });
2561
2561
2562     });
2563
2564     });
2565
2565
2566     });
2567
2568     });
2569
2569
2570     });
2571
2572     });
2573
2573
2574     });
2575
2576     });
2577
2577
2578     });
2579
2580     });
2581
2581
2582     });
2583
2584     });
2585
2585
2586     });
2587
2588     });
2589
2589
2590     });
2591
2592     });
2593
2593
2594     });
2595
2596     });
2597
2597
2598     });
2599
2600     });
2601
2601
2602     });
2603
2604     });
2605
2605
2606     });
2607
2608     });
2609
2609
2610     });
2611
2612     });
2613
2613
2614     });
2615
2616     });
2617
2617
2618     });
2619
2620     });
2621
2621
2622     });
2623
2624     });
2625
2625
2626     });
2627
2628     });
2629
2629
2630     });
2631
2632     });
2633
2633
2634     });
2635
2636     });
2637
2637
2638     });
2639
2640     });
2641
2641
2642     });
2643
2644     });
2645
2645
2646     });
2647
2648     });
2649
2649
2650     });
2651
2652     });
2653
2653
2654     });
2655
2656     });
2657
2657
2658     });
2659
2660     });
2661
2661
2662     });
2663
2664     });
2665
2665
2666     });
2667
2668     });
2669
2669
2670     });
2671
2672     });
2673
2673
2674     });
2675
2676     });
2677
2677
2678     });
2679
2680     });
2681
2681
2682     });
2683
2684     });
2685
2685
2686     });
2687
2688     });
2689
2689
2690     });
2691
2692     });
2693
2693
2694     });
2695
2696     });
2697
2697
2698     });
2699
2700     });
2701
2701
2702     });
2703
2704     });
2705
2705
2706     });
2707
2708     });
2709
2709
2710     });
2711
2712     });
2713
2713
2714     });
2715
2716     });
2717
2717
2718     });
2719
2720     });
2721
2721
2722     });
2723
2724     });
2725
2725
2726     });
2727
2728     });
2729
2729
2730     });
2731
2732     });
2733
2733
2734     });
2735
2736     });
2737
2737
2738     });
2739
2740     });
2741
2741
2742     });
2743
2744     });
2745
2745
2746     });
2747
2748     });
2749
2749
2750     });
2751
2752     });
2753
2753
2754     });
2755
2756     });
2757
2757
2758     });
2759
2760     });
2761
2761
2762     });
2763
2764     });
2765
2765
2766     });
2767
2768     });
2769
2769
2770     });
2771
2772     });
2773
2773
2774     });
2775
2776     });
2777
2777
2778     });
2779
2780     });
2781
2781
2782     });
2783
2784     });
2785
2785
2786     });
2787
2788     });
2789
2789
2790     });
2791
2792     });
2793
2793
2794     });
2795
2796     });
2797
2797
2798     });
2799
2800     });
2801
2801
2802     });
2803
2804     });
2805
2805
2806     });
2807
2808     });
2809
2809
2810     });
2811
2812     });
2813
2813
2814     });
2815
2816     });
2817
2817
2818     });
2819
2820     });
2821
2821
2822     });
2823
2824     });
2825
2825
2826     });
2827
2828     });
2829
2829
2830     });
2831
2832     });
2833
2833
2834     });
2835
2836     });
2837
2837
2838     });
2839
2840     });
2841
2841
2842     });
2843
2844     });
2845
2845
2846     });
2847
2848     });
2849
2849
2850     });
2851
2852     });
2853
2853
2854     });
2855
2856     });
2857
2857
2858     });
2859
2860     });
2861
2861
2862     });
2863
2864     });
2865
2865
2866     });
2867
2868     });
2869
2869
2870     });
2871
2872     });
2873
2873
2874     });
2875
2876     });
2877
2877
2878     });
2879
2880     });
2881
2881
2882     });
2883
2884     });
2885
2885
2886     });
2887
2888     });
2889
2889
2890     });
2891
2892     });
2893
2893
2894     });
2895
2896     });
2897
2897
2898     });
2899
2900     });
2901
2901
2902     });
2903
2904     });
2905
2905
2906     });
2907
2908     });
2909
2909
2910     });
2911
2912     });
2913
2913
2914     });
2915
2916     });
2917
2917
2918     });
2919
2920     });
2921
2921
2922     });
2923
2924     });
2925
2925
2926     });
2927
2928     });
2929
2929
2930     });
2931
2932     });
2933
2933
2934     });
2935
2936     });
2937
2937
2938     });
2939
2940     });
2941
2941
2942     });
2943
2944     });
2945
2945
2946     });
2947
2948     });
2949
2949
2950     });
2951
2952     });
2953
2953
2954     });
2955
2956     });
2957
2957
2958     });
2959
2960     });
2961
2961
2962     });
2963
2964     });
2965
2965
2966     });
2967
2968     });
2969
2969
2970     });
2971
2972     });
2973
2973
2974     });
2975
2976     });
2977
2977
2978     });
2979
2980     });
2981
2981
2982     });
2983
2984     });
2985
2985
2986     });
2987
2988     });
2989
2989
2990     });
2991
2992     });
2993
2993
2994     });
2995
2996     });
2997
2997
2998     });
2999
3000     });
3001
3001
3002     });
3003
3004     });
3005
3005
3006     });
3007
3008     });
3009
3009
3010     });
3011
3012     });
3013
3013
3014     });
3015
3016     });
3017
3017
3018     });
3019
3020     });
3021
3021
3022     });
3023
3024     });
3025
3025
3026     });
3027
3028     });
3029
3029
3030     });
3031
3032     });
3033
3033
3034     });
3035
3036     });
3037
3037
3038     });
3039
3040     });
3041
3041
3042     });
3043
3044     });
3045
3045
3046     });
3047
3048     });
3049
3049
3050     });
3051
3052     });
3053
3053
3054     });
3055
3056     });
3057
3057
3058     });
3059
3060     });
3061
3061
3062     });
3063
3064     });
3065
3065
3066     });
3067
3068     });
3069
3069
3070     });
3071
3072     });
3073
3073
3074     });
3075
3076     });
3077
3077
3078     });
3079
3080     });
3081
3081
3082     });
3083
3084     });
3085
3085
3086     });
3087
3088     });
3089
3089
3090     });
3091
3092     });
3093
3093
3094     });
3095
3096     });
3097
3097
3098     });
3099
3100     });
3101
3101
3102     });
3103
3104     });
3105
3105
3106     });
3107
3108     });
3109
3109
3110     });
3111
3112     });
3113
3113
3114     });
3115
3116     });
3117
3117
3118     });
3119
3120     });
3121
3121
3122     });
3123
3124     });
3125
3125
3126     });
3127
3128     });
3129
3129
3130     });
3131
31
```

## realtor.js

```
1285     if ($("#txtnickname").val().trim().length <= 3) {
1286         navigator.notification.alert(mylabel("Nickname troppo corto"), function() {
1287             });
1288     } else {
1289         account.checknick(account.data.nickname, account.data.emailuser,
1290             function(result) {
1291                 //successo
1292                 console.log("risultato checknick:" + result);
1293                 if (result == "false") {
1294                     account.data.codein = $("#txtcode").val().trim();
1295                     account.check();
1296                     // controllo validità e aggiorno stato bottone
1297                     account.save();
1298                     // salvo su storage locale account
1299                     //navigator.notification.alert(mylabel("Account creato"), function() {});
1300                     $.mobile.changePage($("#homePage"), {
1301                         transition : "slideup",
1302                         changeHash : false
1303                     });
1304                 } else {
1305                     navigator.notification.alert(mylabel("Nickname gia esiste"), function() {
1306                         });
1307                     }, function() {
1308                         });
1309                     //errore
1310                 };
1311             } else {
1312                 navigator.notification.alert(mylabel("Codice non valido"), function() {
1313                     });
1314             };
1315         };
1316     };
1317 } else {
1318     console.log('Connessione Internet non disponibile');
1319     navigator.notification.alert(mylabel("Connessione Internet non disponibile!"),
1320         function() {
1321             });
1322         );
1323     });
1324     //$(".slider").slider();
1325     $("select").selectmenu();
```

## realtor.js

```
1326 //inizializzo prima le select per permettermi di refreshare
1327 //svuoto pagina prima del caricamento scheda
1328 $("#btnAddImmob").on("tap", function() {
1329
1330     //console.log("inizializzo inserminto pagina scheda");
1331     $(".datischeda:text").val("");
1332
1333     $(".datischeda").each(function() {
1334         if ($(this).attr('type') == "number") {
1335             $(this).val("0");
1336         }
1337     });
1338
1339     //$( ".datischeda.slider" ).slider( "option", "value", "0" );
1340     //$( "#elevator" ).slider( "option", "value", "0" );
1341     $("#currency").val("€");
1342
1343     $("#gallery").empty();
1344     scheda.data.ID = 0;
1345     //identifico nuovo inserimento
1346     scheda.data.link = 0;
1347     //identifico nuovo inserimento
1348     $.mobile.changePage($("#scheda"), {
1349         transition : "slideup",
1350         changeHash : true,
1351         reloadPage : true
1352     });
1353
1354 });
1355
1356 $("#scheda").on("pageshow", function(event) {
1357
1358     //console.log("evento inserimento pagina scheda");
1359     $('select').selectmenu('refresh', true);
1360     if ((scheda.data.ID > 0)) {
1361         $("#infoID").show();
1362     } else {
1363         $("#infoID").hide();
1364     }
1365
1366     console.log("id:" + scheda.data.ID);
1367     if ((scheda.data.ID == 0) || ((scheda.data.nickname == account.data.nickname) &&
(account.data.codein == account.data.codeout ) && (account.data.codeout != "") ) &&
```

## realtor.js

```
(account.data.codeout != undefined))) {  
1368     //$("#.datischeda").prop("readonly",false);  
1369     console.log("scheda modificabile");  
1370     $('.datischeda').removeAttr('disabled');  
1371     if (app.isOnline) {  
1372         $("#map-search").show();  
1373     } else {  
1374         $("#map-search").hide();  
1375     }  
1376     $("#btnAlbum").show();  
1377     $("#btnFoto").show();  
1378     $("#btnSalva").show();  
1379 } else {  
1380     //$("#.datischeda").prop("readonly",true);  
1381     console.log("scheda non modificabile");  
1382     $('.datischeda').attr('disabled', 'disabled');  
1383     $("#map-search").hide();  
1384     $("#btnAlbum").hide();  
1385     $("#btnFoto").hide();  
1386     $("#btnSalva").hide();  
1387 }  
1388 /*  
1390 var tabindex = 1;  
1391 $(".datischeda").each(function() {  
1392     console.log("id:"+this.id);  
1393     if (this.type != "hidden") {  
1394         var $input = $(this);  
1395         $input.attr("tabindex", tabindex);  
1396         tabindex++;  
1397     }  
1398 });  
1399 */  
1400 //$('.slider').slider('refresh');  
1401});  
1402 $("#infoHome").on("tap", function(event) {  
1403  
1404     navigator.notification.confirm(mylabel("Richiesta contatto per informazione su  
Realtor"), function(buttonIndex) {  
1405         if (buttonIndex == 1) {  
1406             var msg = 'richiesta conatatto da:' + account.data.nickname + " email:" +  
account.data.emailuser;  
1407             console.log(msg);  
1408         }  
1409     });  
1410 });  
1411});  
1412});  
1413});  
1414});  
1415});  
1416});  
1417});  
1418});  
1419});  
1420});  
1421});  
1422});  
1423});  
1424});  
1425});  
1426});  
1427});  
1428});  
1429});  
1430});  
1431});  
1432});  
1433});  
1434});  
1435});  
1436});  
1437});  
1438});  
1439});  
1440});  
1441});  
1442});  
1443});  
1444});  
1445});  
1446});  
1447});  
1448});  
1449});  
1450});  
1451});  
1452});  
1453});  
1454});  
1455});  
1456});  
1457});  
1458});  
1459});  
1460});  
1461});  
1462});  
1463});  
1464});  
1465});  
1466});  
1467});  
1468});  
1469});  
1470});  
1471});  
1472});  
1473});  
1474});  
1475});  
1476});  
1477});  
1478});  
1479});  
1480});  
1481});  
1482});  
1483});  
1484});  
1485});  
1486});  
1487});  
1488});  
1489});  
1490});  
1491});  
1492});  
1493});  
1494});  
1495});  
1496});  
1497});  
1498});  
1499});  
1500});  
1501});  
1502});  
1503});  
1504});  
1505});  
1506});  
1507});  
1508});  
1509});  
1510});  
1511});  
1512});  
1513});  
1514});  
1515});  
1516});  
1517});  
1518});  
1519});  
1520});  
1521});  
1522});  
1523});  
1524});  
1525});  
1526});  
1527});  
1528});  
1529});  
1530});  
1531});  
1532});  
1533});  
1534});  
1535});  
1536});  
1537});  
1538});  
1539});  
1540});  
1541});  
1542});  
1543});  
1544});  
1545});  
1546});  
1547});  
1548});  
1549});  
1550});  
1551});  
1552});  
1553});  
1554});  
1555});  
1556});  
1557});  
1558});  
1559});  
1560});  
1561});  
1562});  
1563});  
1564});  
1565});  
1566});  
1567});  
1568});  
1569});  
1570});  
1571});  
1572});  
1573});  
1574});  
1575});  
1576});  
1577});  
1578});  
1579});  
1580});  
1581});  
1582});  
1583});  
1584});  
1585});  
1586});  
1587});  
1588});  
1589});  
1590});  
1591});  
1592});  
1593});  
1594});  
1595});  
1596});  
1597});  
1598});  
1599});  
1600});  
1601});  
1602});  
1603});  
1604});  
1605});  
1606});  
1607});  
1608});  
1609});  
1610});  
1611});  
1612});  
1613});  
1614});  
1615});  
1616});  
1617});  
1618});  
1619});  
1620});  
1621});  
1622});  
1623});  
1624});  
1625});  
1626});  
1627});  
1628});  
1629});  
1630});  
1631});  
1632});  
1633});  
1634});  
1635});  
1636});  
1637});  
1638});  
1639});  
1640});  
1641});  
1642});  
1643});  
1644});  
1645});  
1646});  
1647});  
1648});  
1649});  
1650});  
1651});  
1652});  
1653});  
1654});  
1655});  
1656});  
1657});  
1658});  
1659});  
1660});  
1661});  
1662});  
1663});  
1664});  
1665});  
1666});  
1667});  
1668});  
1669});  
1670});  
1671});  
1672});  
1673});  
1674});  
1675});  
1676});  
1677});  
1678});  
1679});  
1680});  
1681});  
1682});  
1683});  
1684});  
1685});  
1686});  
1687});  
1688});  
1689});  
1690});  
1691});  
1692});  
1693});  
1694});  
1695});  
1696});  
1697});  
1698});  
1699});  
1700});  
1701});  
1702});  
1703});  
1704});  
1705});  
1706});  
1707});  
1708});  
1709});  
1710});  
1711});  
1712});  
1713});  
1714});  
1715});  
1716});  
1717});  
1718});  
1719});  
1720});  
1721});  
1722});  
1723});  
1724});  
1725});  
1726});  
1727});  
1728});  
1729});  
1730});  
1731});  
1732});  
1733});  
1734});  
1735});  
1736});  
1737});  
1738});  
1739});  
1740});  
1741});  
1742});  
1743});  
1744});  
1745});  
1746});  
1747});  
1748});  
1749});  
1750});  
1751});  
1752});  
1753});  
1754});  
1755});  
1756});  
1757});  
1758});  
1759});  
1760});  
1761});  
1762});  
1763});  
1764});  
1765});  
1766});  
1767});  
1768});  
1769});  
1770});  
1771});  
1772});  
1773});  
1774});  
1775});  
1776});  
1777});  
1778});  
1779});  
1780});  
1781});  
1782});  
1783});  
1784});  
1785});  
1786});  
1787});  
1788});  
1789});  
1790});  
1791});  
1792});  
1793});  
1794});  
1795});  
1796});  
1797});  
1798});  
1799});  
1800});  
1801});  
1802});  
1803});  
1804});  
1805});  
1806});  
1807});  
1808});  
1809});  
1810});  
1811});  
1812});  
1813});  
1814});  
1815});  
1816});  
1817});  
1818});  
1819});  
1820});  
1821});  
1822});  
1823});  
1824});  
1825});  
1826});  
1827});  
1828});  
1829});  
1830});  
1831});  
1832});  
1833});  
1834});  
1835});  
1836});  
1837});  
1838});  
1839});  
1840});  
1841});  
1842});  
1843});  
1844});  
1845});  
1846});  
1847});  
1848});  
1849});  
1850});  
1851});  
1852});  
1853});  
1854});  
1855});  
1856});  
1857});  
1858});  
1859});  
1860});  
1861});  
1862});  
1863});  
1864});  
1865});  
1866});  
1867});  
1868});  
1869});  
1870});  
1871});  
1872});  
1873});  
1874});  
1875});  
1876});  
1877});  
1878});  
1879});  
1880});  
1881});  
1882});  
1883});  
1884});  
1885});  
1886});  
1887});  
1888});  
1889});  
1890});  
1891});  
1892});  
1893});  
1894});  
1895});  
1896});  
1897});  
1898});  
1899});  
1900});  
1901});  
1902});  
1903});  
1904});  
1905});  
1906});  
1907});  
1908});  
1909});  
1910});  
1911});  
1912});  
1913});  
1914});  
1915});  
1916});  
1917});  
1918});  
1919});  
1920});  
1921});  
1922});  
1923});  
1924});  
1925});  
1926});  
1927});  
1928});  
1929});  
1930});  
1931});  
1932});  
1933});  
1934});  
1935});  
1936});  
1937});  
1938});  
1939});  
1940});  
1941});  
1942});  
1943});  
1944});  
1945});  
1946});  
1947});  
1948});  
1949});  
1950});  
1951});  
1952});  
1953});  
1954});  
1955});  
1956});  
1957});  
1958});  
1959});  
1960});  
1961});  
1962});  
1963});  
1964});  
1965});  
1966});  
1967});  
1968});  
1969});  
1970});  
1971});  
1972});  
1973});  
1974});  
1975});  
1976});  
1977});  
1978});  
1979});  
1980});  
1981});  
1982});  
1983});  
1984});  
1985});  
1986});  
1987});  
1988});  
1989});  
1990});  
1991});  
1992});  
1993});  
1994});  
1995});  
1996});  
1997});  
1998});  
1999});  
2000});  
2001});  
2002});  
2003});  
2004});  
2005});  
2006});  
2007});  
2008});  
2009});  
2010});  
2011});  
2012});  
2013});  
2014});  
2015});  
2016});  
2017});  
2018});  
2019});  
2020});  
2021});  
2022});  
2023});  
2024});  
2025});  
2026});  
2027});  
2028});  
2029});  
2030});  
2031});  
2032});  
2033});  
2034});  
2035});  
2036});  
2037});  
2038});  
2039});  
2040});  
2041});  
2042});  
2043});  
2044});  
2045});  
2046});  
2047});  
2048});  
2049});  
2050});  
2051});  
2052});  
2053});  
2054});  
2055});  
2056});  
2057});  
2058});  
2059});  
2060});  
2061});  
2062});  
2063});  
2064});  
2065});  
2066});  
2067});  
2068});  
2069});  
2070});  
2071});  
2072});  
2073});  
2074});  
2075});  
2076});  
2077});  
2078});  
2079});  
2080});  
2081});  
2082});  
2083});  
2084});  
2085});  
2086});  
2087});  
2088});  
2089});  
2090});  
2091});  
2092});  
2093});  
2094});  
2095});  
2096});  
2097});  
2098});  
2099});  
2100});  
2101});  
2102});  
2103});  
2104});  
2105});  
2106});  
2107});  
2108});  
2109});  
2110});  
2111});  
2112});  
2113});  
2114});  
2115});  
2116});  
2117});  
2118});  
2119});  
2120});  
2121});  
2122});  
2123});  
2124});  
2125});  
2126});  
2127});  
2128});  
2129});  
2130});  
2131});  
2132});  
2133});  
2134});  
2135});  
2136});  
2137});  
2138});  
2139});  
2140});  
2141});  
2142});  
2143});  
2144});  
2145});  
2146});  
2147});  
2148});  
2149});  
2150});  
2151});  
2152});  
2153});  
2154});  
2155});  
2156});  
2157});  
2158});  
2159});  
2160});  
2161});  
2162});  
2163});  
2164});  
2165});  
2166});  
2167});  
2168});  
2169});  
2170});  
2171});  
2172});  
2173});  
2174});  
2175});  
2176});  
2177});  
2178});  
2179});  
2180});  
2181});  
2182});  
2183});  
2184});  
2185});  
2186});  
2187});  
2188});  
2189});  
2190});  
2191});  
2192});  
2193});  
2194});  
2195});  
2196});  
2197});  
2198});  
2199});  
2200});  
2201});  
2202});  
2203});  
2204});  
2205});  
2206});  
2207});  
2208});  
2209});  
2210});  
2211});  
2212});  
2213});  
2214});  
2215});  
2216});  
2217});  
2218});  
2219});  
2220});  
2221});  
2222});  
2223});  
2224});  
2225});  
2226});  
2227});  
2228});  
2229});  
2230});  
2231});  
2232});  
2233});  
2234});  
2235});  
2236});  
2237});  
2238});  
2239});  
2240});  
2241});  
2242});  
2243});  
2244});  
2245});  
2246});  
2247});  
2248});  
2249});  
2250});  
2251});  
2252});  
2253});  
2254});  
2255});  
2256});  
2257});  
2258});  
2259});  
2260});  
2261});  
2262});  
2263});  
2264});  
2265});  
2266});  
2267});  
2268});  
2269});  
2270});  
2271});  
2272});  
2273});  
2274});  
2275});  
2276});  
2277});  
2278});  
2279});  
2280});  
2281});  
2282});  
2283});  
2284});  
2285});  
2286});  
2287});  
2288});  
2289});  
2290});  
2291});  
2292});  
2293});  
2294});  
2295});  
2296});  
2297});  
2298});  
2299});  
2300});  
2301});  
2302});  
2303});  
2304});  
2305});  
2306});  
2307});  
2308});  
2309});  
2310});  
2311});  
2312});  
2313});  
2314});  
2315});  
2316});  
2317});  
2318});  
2319});  
2320});  
2321});  
2322});  
2323});  
2324});  
2325});  
2326});  
2327});  
2328});  
2329});  
2330});  
2331});  
2332});  
2333});  
2334});  
2335});  
2336});  
2337});  
2338});  
2339});  
2340});  
2341});  
2342});  
2343});  
2344});  
2345});  
2346});  
2347});  
2348});  
2349});  
2350});  
2351});  
2352});  
2353});  
2354});  
2355});  
2356});  
2357});  
2358});  
2359});  
2360});  
2361});  
2362});  
2363});  
2364});  
2365});  
2366});  
2367});  
2368});  
2369});  
2370});  
2371});  
2372});  
2373});  
2374});  
2375});  
2376});  
2377});  
2378});  
2379});  
2380});  
2381});  
2382});  
2383});  
2384});  
2385});  
2386});  
2387});  
2388});  
2389});  
2390});  
2391});  
2392});  
2393});  
2394});  
2395});  
2396});  
2397});  
2398});  
2399});  
2400});  
2401});  
2402});  
2403});  
2404});  
2405});  
2406});  
2407});  
2408});  
2409});  
2410});  
2411});  
2412});  
2413});  
2414});  
2415});  
2416});  
2417});  
2418});  
2419});  
2420});  
2421});  
2422});  
2423});  
2424});  
2425});  
2426});  
2427});  
2428});  
2429});  
2430});  
2431});  
2432});  
2433});  
2434});  
2435});  
2436});  
2437});  
2438});  
2439});  
2440});  
2441});  
2442});  
2443});  
2444});  
2445});  
2446});  
2447});  
2448});  
2449});  
2450});  
2451});  
2452});  
2453});  
2454});  
2455});  
2456});  
2457});  
2458});  
2459});  
2460});  
2461});  
2462});  
2463});  
2464});  
2465});  
2466});  
2467});  
2468});  
2469});  
2470});  
2471});  
2472});  
2473});  
2474});  
2475});  
2476});  
2477});  
2478});  
2479});  
2480});  
2481});  
2482});  
2483});  
2484});  
2485});  
2486});  
2487});  
2488});  
2489});  
2490});  
2491});  
2492});  
2493});  
2494});  
2495});  
2496});  
2497});  
2498});  
2499});  
2500});  
2501});  
2502});  
2503});  
2504});  
2505});  
2506});  
2507});  
2508});  
2509});  
2510});  
2511});  
2512});  
2513});  
2514});  
2515});  
2516});  
2517});  
2518});  
2519});  
2520});  
2521});  
2522});  
2523});  
2524});  
2525});  
2526});  
2527});  
2528});  
2529});  
2530});  
2531});  
2532});  
2533});  
2534});  
2535});  
2536});  
2537});  
2538});  
2539});  
2540});  
2541});  
2542});  
2543});  
2544});  
2545});  
2546});  
2547});  
2548});  
2549});  
2550});  
2551});  
2552});  
2553});  
2554});  
2555});  
2556});  
2557});  
2558});  
2559});  
2560});  
2561});  
2562});  
2563});  
2564});  
2565});  
2566});  
2567});  
2568});  
2569});  
2570});  
2571});  
2572});  
2573});  
2574});  
2575});  
2576});  
2577});  
2578});  
2579});  
2580});  
2581});  
2582});  
2583});  
2584});  
2585});  
2586});  
2587});  
2588});  
2589});  
2590});  
2591});  
2592});  
2593});  
2594});  
2595});  
2596});  
2597});  
2598});  
2599});  
2600});  
2601});  
2602});  
2603});  
2604});  
2605});  
2606});  
2607});  
2608});  
2609});  
2610});  
2611});  
2612});  
2613});  
2614});  
2615});  
2616});  
2617});  
2618});  
2619});  
2620});  
2621});  
2622});  
2623});  
2624});  
2625});  
2626});  
2627});  
2628});  
2629});  
2630});  
2631});  
2632});  
2633});  
2634});  
2635});  
2636});  
2637});  
2638});  
2639});  
2640});  
2641});  
2642});  
2643});  
2644});  
2645});  
2646});  
2647});  
2648});  
2649});  
2650});  
2651});  
2652});  
2653});  
2654});  
2655});  
2656});  
2657});  
2658});  
2659});  
2660});  
2661});  
2662});  
2663});  
2664});  
2665});  
2666});  
2667});  
2668});  
2669});  
2670});  
2671});  
2672});  
2673});  
2674});  
2675});  
2676});  
2677});  
2678});  
2679});  
2680});  
2681});  
2682});  
2683});  
2684});  
2685});  
2686});  
2687});  
2688});  
2689});  
2690});  
2691});  
2692});  
2693});  
2694});  
2695});  
2696});  
2697});  
2698});  
2699});  
2700});  
2701});  
2702});  
2703});  
2704});  
2705});  
2706});  
2707});  
2708});  
2709});  
2710});  
2711});  
2712});  
2713});  
2714});  
2715});  
2716});  
2717});  
2718});  
2719});  
2720});  
2721});  
2722});  
27
```

## realtor.js

```
1408     mail.data.to = "realtor@dntppe.net";
1409     mail.data.subject = "Info Generic";
1410     mail.data.body = msg;
1411     value = JSON.stringify(mail.data);
1412     console.log(value);
1413     mail.send(value, function() {
1414         console.log('Email inviated');
1415
1416         navigator.notification.alert(mylabel("Email inviated"), function() {
1417             });
1418         }, function() {
1419             //nel caso di errore
1420             console.log('Email not inviated');
1421             navigator.notification.alert(mylabel("Email not inviated"), function() {
1422                 });
1423             });
1424         );
1425     );
1426 );
1427
1428 $("#infoID").on("tap", function(event) {
1429
1430     navigator.notification.confirm(mylabel("Richiesta contatto per informazione su questa
1431     scheda"), function(buttonIndex) {
1432         if(buttonIndex == 1) {
1433             var database = "DB_" + scheda.data.country.toUpperCase().trim() + "_" +
1434             scheda.data.administrative_area_level_2.toUpperCase().trim();
1435             var msg = 'richiesta conatatto da:' + account.data.nickname + " email:" +
1436             account.data.emailuser + " db:" + database + " id:" + scheda.data.ID + " link:" +
1437             scheda.data.link;
1438             console.log(msg);
1439             mail.data.to = "realtor@dntppe.net";
1440             mail.data.subject = "Info ID";
1441             mail.data.body = msg;
1442             value = JSON.stringify(mail.data);
1443             console.log(value);
1444             mail.send(value, function() {
1445                 console.log('Email inviated');
1446
1447                 navigator.notification.alert(mylabel("Email inviated"), function() {
1448                     });
1449                 }, function() {
1450                     //nel caso di errore
```

realtor.js

```

1447         console.log('Email not inviated');
1448         navigator.notification.alert(mylabel("Email not inviated"), function() {
1449             });
1450         });
1451     );
1452   );
1453 );
1454 //creca posizione da gps
1455 $("#btnPosizione").on("tap", function() {
1456     geo.posizione(scheda.onPositionSuccess);
1457 });
1458 //creca posizione da gps
1459 $("#btnSearch").on("tap", function() {
1460     if (app.isOnline()) {
1461         console.log('Connessione Internet disponibile cerco indirizzo:' + $
1462         ("#txtIndirizzofind").val().trim());
1463         scheda.cercaindirizzo($("#txtIndirizzofind").val().trim());
1464     } else {
1465         console.log('Connessione Internet non disponibile');
1466         navigator.notification.alert(mylabel("Connessione Internet non disponibile!"));
1467         function() {
1468             });
1469     });
1470 //assegnamento della gestione dell'evento tap, l'evento corrispondente al tocco sul
1471 //pulsante
1472     $("#btnSalva").on("tap", function() {
1473         if (scheda.data["ID"] > 0) {
1474             navigator.notification.confirm(mylabel("Vuoi modificare questa scheda?"),
1475             function(buttonIndex) {
1476                 if (buttonIndex == 1) {
1477                     console.log("modifico una mia scheda");
1478                 } else {
1479                     scheda.data["link"] = scheda.data["ID"];
1480                     //creo una copia e tengo traccia in db
1481                     console.log("sono una nuova scheda");
1482                 }
1483             });
1484             scheda.save();
1485         });

```

## realtor.js

```
1486
1487     } else {
1488         scheda.save();
1489     }
1490
1491 });
1492
1493 //Al pulsante exit assegniamo il metodo per chiudere l'applicazione
1494 $("#btnExit").on("tap", app.exit);
1495
1496 //elenco delle schede memorizzate uso l'evento pagebeforeshow della schermata
1497 elencoSchede
1498 $("#elencoSchede").on("pagebeforeshow", function(event) {
1499
1500     var elencoSchede = $("#liElencoSchede");
1501     //id in index.html
1502     elencoSchede.html("");
1503     //svuoto
1504
1505     var li = $("<li data-role='list-divider' role='heading>" + mylabel('scheda') + "</li>");
1506     elencoSchede.append(li);
1507
1508     // per tutti gli elementi memorizzati
1509     for (var i = 0; i < app.storage.length; i++) {
1510         if (app.storage.key(i) != 'user') {
1511             //per ciascun elemento creo un item
1512             console.log('Carico da storage e creo elemento lista ' + i + ' su ' +
1513             app.storage.length);
1514
1515             var li = $("<li data-theme='c'><a href='#' data-transition='slide'>" +
1516             app.storage.key(i) + //metodo key() consente di recuperare la chiave dell'i-esimo elemento
1517             "</a></li>");
1518
1519             li.on("tap", function() {
1520                 var key = $(this).text();
1521                 console.log('Per visualizzare carico dati elemento lista ' + key);
1522                 scheda.load(key);
1523                 app.caricoscheda();
1524             });
1525
1526             //swipeleft sull'elemento della listview
1527             li.on("swipeleft", function() {
1528                 var key = $(this);
```

## realtor.js

```
1526     console.log('pressione prolungata rimuovo scheda ' + key.text());
1527     navigator.notification.confirm("//chiedo conferma
1528     mylabel("Confermi l'eliminazione della scheda?"), function(buttonIndex) {
1529         if (buttonIndex == 1) {
1530             scheda.remove(key.text());
1531             console.log("Aggiorno pagina dopo rimozione scheda");
1532             key.remove();
1533             $("#liElencoSchede").listview("refresh");
1534         }
1535     });
1536
1537     });
1538
1539     elencoSchede.append(li);
1540     //appendo l'item
1541     }
1542     ;
1543     elencoSchede.listview("refresh");
1544 );
1545
1546 //creca posizione da gps
1547 $("#btnSearch2").on("tap", function() {
1548     if (app.isOnline()) {
1549         var input = $("#txtIndirizzofind2").val().trim();
1550
1551         console.log('Connessione Internet disponibile cerco indirizzo:' + $
1552         ("#txtIndirizzofind2").val().trim());
1553         console.log("Cerco indirizzo: " + input);
1554         geo.search({
1555             'address' : input
1556         }, function(address, longitude, latitude) {
1557             $(address).each(function() {
1558                 console.log(this.types[0] + ":" + this.long_name);
1559             });
1560             app.mostramappa(longitude, latitude);
1561         });
1562     } else {
1563         console.log('Connessione Internet non disponibile');
1564         navigator.notification.alert(mylabel("Connessione Internet non disponibile")),
1565         function() {
1566             });
1566     }
```

## realtor.js

```
1567      });
1568      //creca posizione da gps
1569      $("#btnPosizione2").on("tap", function() {
1570          geo.posizione(function(position) {
1571              app.mostramappa(position.coords.longitude, position.coords.latitude);
1572          });
1573      });
1574
1575      //elenco delle tabelle memorizzate uso l'evento pagebeforeshow della schermata
1576      elencoTabelle
1577      $("#elencoTabelle").on("pagebeforeshow", function(event) {
1578          app.showdb(app.tabella.lowlimit, app.tabella.highlimit, function(input) {//success
1579              var elenco = $("#liElencoTable");
1580              //id in index.html
1581              elenco.html("");
1582              //svuoto
1583              app.schede.selezionate = [];
1584              // var li = $("<li data-role='list-divider' role='heading'>" + mylabel('tabella') + "</li>");
1585              // elenco.append(li);
1586              for (var i = 0; i < input.length; i++) {
1587                  var li = $("<li data-theme='c'><a href='#' data-transition='slide'>" + input[i] + "</a></li>");
1588                  li.on("tap", function() {
1589                      app.tabella.nome = $(this).text();
1590                      app.tabella.coord = [-180, 180, -180, 180];
1591                      $.mobile.changePage($("#elencoSchedeServer"), {
1592                          transition : "slideup",
1593                          changeHash : true
1594                      });
1595                      elenco.append(li);
1596                  };
1597                  elenco.listview("refresh");
1598              }, function() {//error
1599
1600          });
1601      });
1602      //elenco schede memorizzate sul server uso l'evento pagebeforeshow della schermata
1603      elencoTabelle
1604      $("#elencoSchedeServer").on("pagebeforeshow", function(event) {
1605          var richiesta = JSON.stringify(app.tabella);
1606          app.showschede(richiesta, function(input) {//success
1607              console.log(JSON.stringify(input));
1608          });
1609      });
1610  
```

## realtor.js

```
1607      var elenco = $("#lielencoSchedeServer");
1608      //id in index.html
1609      elenco.html("");
1610      //svuoto
1611
1612      // var li = $("<li data-role='list-divider' role='heading'>" + mylabel('scheda') + "</
1613      li>");
1614      // elenco.append(li);
1615
1616      for (var i = 0; i < input.length; i++) {
1617          var li = $("<li data-icon='false' id='" + app.tabella.nome + '/' + input[i][0] + "'><a
1618          href='#' data-transition='slide' data-theme='c'>" + input[i][0] + ':' + input[i][1] + " " + input[i]
1619          [2] + " " + input[i][3] + " " + input[i][4] + "</a></li>");
1620          li.on("swiperight", app.schedadaserver);
1621          li.on("tap", function() {
1622
1623              console.log("tap elemento lista : " + $(this).attr('id'));
1624              var pos = app.schede.selezionate.indexOf($(this).attr('id'));
1625              console.log("posizione." + pos);
1626              if (pos >= 0) {
1627                  $(this).buttonMarkup({
1628                      icon : "",
1629                      iconpos : "right",
1630                      shadow : "false"
1631                  });
1632                  app.schede.selezionate.splice(pos, 1);
1633              } else {
1634                  $(this).buttonMarkup({
1635                      icon : "check",
1636                      iconpos : "right",
1637                      shadow : "false"
1638                  });
1639              });
1640
1641          var pos = app.schede.selezionate.indexOf(li.attr('id'));
1642          if (pos < 0) {
1643              li.buttonMarkup({
1644                  icon : "",
1645                  iconpos : "right",
1646                  shadow : "false"
```

## realtor.js

```
1647      });
1648  } else {
1649      li.buttonMarkup({
1650          icon : "check",
1651          iconpos : "right",
1652          shadow : "false"
1653      });
1654  };
1655  elenco.append(li);
1656
1657  };
1658  elenco.listView("refresh");
1659
1660  }, function() {
1661  });
1662 //error
1663
1664 });
1665
1666 //creazione tabella statistiche
1667 $("#btnStat").on("tap", function(event) {
1668
1669     /*
1670     *
1671     *         var elencoSchede = $("#liElencoSchede"); //id in index.html
1672     elencoSchede.html(""); //svuoto
1673
1674     var li = $("<li data-role='list-divider' role='heading'>" + myLabel('scheda') + "</li>");
1675     elencoSchede.append(li);
1676
1677     *
1678     *
1679     */
1680     if (app.schede.selezionate.length > 0) {
1681         var col_tabella = $("#colonnetabella");
1682         //id in index.html
1683         col_tabella.html("");
1684         //svuoto colonne
1685         var colonna = ("<th data-priority='1'>Nr</th>");
1686         colonna = colonna + ("<th data-priority='6'>surface</th>");
1687         colonna = colonna + ("<th data-priority='5'>price</th>");
1688         colonna = colonna + ("<th data-priority='4'>normalized</th>");
1689         colonna = colonna + ("<th data-priority='3'>price/mq</th>");
```

## realtor.js

```
1690    colonna = colonna + ("<th data-priority='2'>price/sup eq</th>");
1691    col_tabella.append(colonna);
1692
1693    var riga_tabella = $("#righetabella");
1694    //id in index.html
1695    riga_tabella.html("");
1696    //svuoto colonne
1697    var i = 1;
1698    $.each(app.schede.selezionate, function(key, value) {
1699        app.schede.nome = value;
1700        console.log("id scheda selezionata:" + app.schede.nome);
1701        app.getscheda(app.schede.nome, function(input) {//success
1702            console.log(key + ": " + value);
1703            //$( ".datischeda#" + key).val(value);
1704            var riga = ("<tr id=\"" + value + "\">");
1705            //inizio riga
1706            //contenuto riga a colonna
1707            riga = riga + ("<th>" + input["ID"] + "</th>");
1708            riga = riga + ("<td>" + input["surface"] + "</td>");
1709            riga = riga + ("<td>" + input["price"] + "</td>");
1710            riga = riga + ("<td>" + input["normalized"] + "</td>");
1711            var pricemq = 0;
1712            if (input["surface"] > 0) {
1713                pricemq = input["price"] / input["surface"];
1714            }
1715            riga = riga + ("<td>" + pricemq.toFixed(0) + "</td>");
1716            var pricenorm = 0;
1717            if (input["normalized"] > 0) {
1718                pricenorm = input["price"] / input["normalized"];
1719            }
1720            riga = riga + ("<td>" + pricenorm.toFixed(0) + "</td>");
1721            riga = riga + ("</tr>");
1722            //fine riga
1723
1724            var rigaq = $(riga);
1725            rigaq.on("tap", app.schedadaserver);
1726            riga_tabella.append(rigaq);
1727            $("#table-column-toggle").table("refresh");
1728        }, function() {//error
1729    });
1730
1731    });
1732
```

## realtor.js

```
1733 //$("#table-column-toggle").trigger("create");
1734 //$("#table-column-toggle").table("refresh");
1735
1736 $.mobile.changePage($("#statistiche"), {
1737     transition : "slideup",
1738     changeHash : true
1739 });
1740 } else {
1741     navigator.notification.alert(mylabel("Nessuna scheda selezionata"), function() {
1742         });
1743     }
1744 });
1745
1746 $("#btnGrafico").on("tap", function(event) {
1747     if (app.schede.selezionate.length > 0) {
1748
1749         // $(function() {
1750         //var width = $('[data-role="page"]').innerWidth();
1751         // var height = $('[data-role="page"]').innerHeight();
1752         var width = $(window).width();
1753         ;
1754         var height = $(window).height();
1755         console.log('width = ' + width);
1756         console.log('height = ' + height);
1757         var min = width;
1758         if (min > height) {
1759             min = height;
1760         }
1761         min = min;
1762         $('#placeholder').width(min - 30);
1763         $('#placeholder').height(min - 40);
1764
1765         function gd(input) {
1766             var parts = input.split('-');
1767             // new Date(year, month [, day [, hours[, minutes[, seconds[, ms]]]]])
1768             return new Date(parts[0], parts[1] - 1, parts[2]).getTime();
1769             // Note: months are 0-based
1770         }
1771
1772         var datasets = {
1773             "price_mq" : {
1774                 label : "price/mq",
1775                 data : []
```

## realtor.js

```
1776          // [gd("1988-02-27"), 483994], [gd("1989-05-20"), 479060],  
1777          [gd("1990-12-10"), 457648], [gd("1991-02-28"), 401949], [gd("1992-12-18"), 424705],  
1778          [gd("1993-08-18"), 402375], [gd("1994-08-18"), 377867], [gd("1995-08-18"), 357382],  
1779          [gd("1996-08-18"), 337946], [gd("1997-08-18"), 336185], [gd("1998-08-18"), 328611],  
1780          [gd("1999-08-18"), 329421], [gd("2000-08-18"), 342172], [gd("2001-08-18"), 344932],  
1781          [gd("2002-08-18"), 387303], [gd("2003-08-18"), 440813], [gd("2004-08-18"), 480451],  
1782          [gd("2005-08-18"), 504638], [gd("2006-08-18"), 528692]]  
1783      },  
1784      "price_norm" : {  
1785          label : "price/sup eq",  
1786          data : []  
1787          // [gd("1988-02-27"), 218000], [gd("1989-05-20"), 203000],  
1788          [gd("1990-12-10"), 171000], [gd("1992-12-18"), 42500], [gd("1993-08-18"), 37600],  
1789          [gd("1994-08-18"), 36600], [gd("1995-08-18"), 21700], [gd("1996-08-18"), 19200],  
1790          [gd("1997-08-18"), 21300], [gd("1998-08-18"), 13600], [gd("1999-08-18"), 14000],  
1791          [gd("2000-08-18"), 19100], [gd("2001-08-18"), 21300], [gd("2002-08-18"), 23600],  
1792          [gd("2003-08-18"), 25100], [gd("2004-08-18"), 26100], [gd("2005-08-18"), 31100],  
1793          [gd("2006-08-18"), 34700]]  
1794      },  
1795      };  
1796      // hard-code color indices to prevent them from shifting as  
1797      // countries are turned on/off  
1798      var i = 0;  
1799      $.each(datasets, function(key, val) {  
1800          val.color = i; ++i;  
1801      });  
1802      // insert checkboxes  
1803      var choiceContainer = $("#choices");  
1804      /* //bug che non riesco a risolvere se non uso html al richiamo della pagina si blocca  
1805      choiceContainer.empty();  
1806      $.each(datasets, function(key, val) {  
1807          choiceContainer.append("<br><input type='checkbox' name='" + key + "'  
1808          checked='checked' id='id" + key + "'></input><label for='id" + key + "'>" + val.label + "</  
1809          label>");  
1810          choiceContainer.append("<label><input type='checkbox' checked='checked'  
1811          name='" + key + "' id='id" + key + "' />" + val.label + "</label>");  
1812      });  
1813      //$("#input[type='checkbox']").checkboxradio("refresh");
```

realtor.js

```
1804 */  
1805  
1806 choiceContainer.find("input").click(function() {  
1807     $("input[type='checkbox']").checkboxradio("refresh");  
1808     plotAccordingToChoices();  
1809 });  
1810  
1811     function plotAccordingToChoices() {  
1812  
1813         var data = [];  
1814  
1815         $("#choices").find("input:checked").each(function() {  
1816             var key = $(this).attr("name");  
1817             if (key && datasets[key]) {  
1818                 data.push(datasets[key]);  
1819             }  
1820         });  
1821  
1822         $("#placeholder").empty();  
1823  
1824         if (data.length > 0) {  
1825  
1826             var plot = $.plot("#placeholder", data, {  
1827                 series : {  
1828                     lines : {  
1829                         show : true  
1830                     },  
1831                     points : {  
1832                         show : true  
1833                     }  
1834                 },  
1835                 yaxis : {  
1836                     position : "right",  
1837                     //show: true,  
1838                     // axisMargin:20  
1839                 },  
1840                 xaxis : {  
1841                     //show: true,  
1842                     mode : "time",  
1843                     timeformat : "%Y/%m",  
1844                     //tickDecimals: 0,  
1845                     //axisMargin:20  
1846             }  
1847         }  
1848     }  
1849 }  
1850  
1851     
```

## realtor.js

```
1847     });
1848
1849     }
1850     plot.setupGrid();
1851     //only necessary if your new data will change the axes or grid
1852     plot.draw();
1853     //$.mobile.changePage($("#grafico"), { changeHash: false});
1854
1855 }
1856
1857
1858     datasets["price_mq"].data = [];
1859     //svuoto array
1860     datasets["price_norm"].data = [];
1861     //svuoto array
1862
1863     $.each(app.schede.selezionate, function(key, value) {
1864         app.schede.nome = value;
1865         console.log("id scheda selezionata:" + app.schede.nome);
1866         app.getscheda(app.schede.nome, function(input) {//success
1867             console.log(key + ": " + value);
1868             var pricemq = 0;
1869             if (input["surface"] > 0) {
1870                 pricemq = input["price"] / input["surface"];
1871             }
1872
1873             var pricenorm = 0;
1874             if (input["normalized"] > 0) {
1875                 pricenorm = input["price"] / input["normalized"];
1876             }
1877
1878             var dato1 = [gd(input["date"]), pricemq];
1879             var dato2 = [gd(input["date"]), pricenorm];
1880             datasets["price_mq"].data.push(dato1);
1881             datasets["price_norm"].data.push(dato2);
1882             datasets["price_mq"].data.sort(function(a, b) {
1883                 return (a[0] - b[0]);
1884             });
1885             datasets["price_norm"].data.sort(function(a, b) {
1886                 return (a[0] - b[0]);
1887             });
1888
1889             console.log(JSON.stringify(datasets));
```

## realtor.js

```
1890     plotAccordingToChoices();
1891
1892         //                      $("#table-column-toggle").table("refresh");
1893     }, function() {//error
1894     });
1895
1896 });
1897 $.mobile.changePage($("#grafico"), {
1898     transition : "slideup",
1899     changeHash : true
1900 });
1901
1902 //});
1903
1904 } else {
1905     navigator.notification.alert(mylabel("Nessuna scheda selezionata"), function() {
1906     });
1907 }
1908
1909 });
1910
1911 $("#btnAll").on("tap", function() {
1912     $("#lielencoSchedeServer li").each(function() {
1913
1914         var pos = app.schede.selezionate.indexOf($(this).attr('id'));
1915         console.log("posizione:" + pos);
1916         if (pos >= 0) {
1917             } else {
1918                 $(this).buttonMarkup({
1919                     icon : "check",
1920                     iconpos : "right",
1921                     shadow : "false"
1922                 });
1923                 app.schede.selezionate.push($(this).attr('id'));
1924             };
1925
1926         });
1927     });
1928
1929 //invio delle schede
1930 $("#btnInviaSchede").on("tap", function() {
1931     if (app.storage.length > 0) {
1932         if (app.isOnline()) {
```

## realtor.js

```
1933         console.log('Connessione Internet disponibile preparo invio schede');
1934         if ((account.data.codein == account.data.codeout) && (account.data.codeout != ""
1935           && (account.data.codeout != undefined)) {
1935             navigator.notification.confirm(mylabel("Confermi l'invio delle schede?"),
1936               function(buttonIndex) {
1937                 if (buttonIndex == 1) {
1938                   console.log('Confermato Invio schede');
1939                   listaSchede = [];
1940
1941                   for (var i = 0; i < app.storage.length; i++) {
1942                     if (app.storage.key(i) != 'user') {
1943                       var nameitem = app.storage.key(i);
1944
1945                       console.log('Memorizzo nome scheda:' + nameitem);
1946                       listaSchede.push(nameitem);
1947                     }
1948                   ;
1949                   console.log('Array item memorizzati:' + listaSchede.length);
1950                   for (var i = 0; i < listaSchede.length; i++) {
1951                     var value = JSON.parse(app.storage.getItem(listaSchede[i]));
1952                     value.nickname = account.data.nickname;
1953                     scheda.addscheda(JSON.stringify(value), function() {
1954                       scheda.remove(value.codename);
1955                     }, function() {
1956                       //nel caso di errore
1957                     });
1958                   };
1959                 );
1960               );
1961             } else {
1962               console.log('Account non creato');
1963               navigator.notification.alert(mylabel("Account non creato"), function() {
1964                 });
1965               );
1966             } else {
1967               console.log('Connessione Internet non disponibile');
1968               navigator.notification.alert(mylabel("Connessione Internet non disponibile!"),
1969                 function() {
1970                   );
1971                 );
1972               } else {
1972                 console.log('Richiesto invio ma nessuna scheda memorizzata');
```

## realtor.js

```
1973      navigator.notification.alert(mylabel("Nessuna scheda memorizzata"), function() {  
1974          });  
1975      };  
1976      });  
1977      });  
1978      );  
1979      $("#btnFoto").on("tap", function() {  
1980          navigator.camera.getPicture(app.onCameraSuccess, app.onCameraError, {  
1981              quality : 75,  
1982              destinationType : Camera.DestinationType.DATA_URL, //FILE_URI,  
1983              sourceType : Camera.PictureSourceType.CAMERA,  
1984              allowEdit : true,  
1985              encodingType : Camera.EncodingType.JPEG,  
1986              targetWidth : 500,  
1987              targetHeight : 500  
1988          });  
1989      });  
1990      );  
1991      $("#btnAlbum").on("tap", function() {  
1992          navigator.camera.getPicture(app.onCameraSuccess, app.onCameraError, {  
1993              quality : 75,  
1994              destinationType : Camera.DestinationType.DATA_URL, //FILE_URI,  
1995              sourceType : Camera.PictureSourceType.SAVEDPHOTOALBUM,  
1996              allowEdit : true,  
1997              encodingType : Camera.EncodingType.JPEG,  
1998              targetWidth : 500,  
1999              targetHeight : 500  
2000          });  
2001      });  
2002      );  
2003      $("#btnAccount").on("tap", function() {  
2004          $.mobile.changePage($("#account"), {  
2005              transition : "slideup",  
2006              changeHash : true  
2007          });  
2008      });  
2009      );  
2010      account.load();  
2011      // carico account da storage come prima partenza e aggiorno icona bottone  
2012      app.receivedEvent('deviceready');  
2013      console.log('deviceready');
```

## realtor.js

```
2016
2017 //eseguo scrypt iniziali in base allo stato online
2018 /* if (app.isOnline()) {
2019     app.online();
2020 } else
2021 {
2022     app.offline();
2023 };*/
2024 document.addEventListener("online", app.online, false);
2025 document.addEventListener("offline", app.offline, false);
2026 },
2027 // Bind Event Listeners
2028 //
2029 // Bind any events that are required on startup. Common events are:
2030 // 'load', 'deviceready', 'offline', and 'online'.
2031 bindEvents : function() {
2032     document.addEventListener('deviceready', app.onDeviceReady, false);
2033 }
2034 },
2035 // Application Constructor
2036 initialize : function() {
2037     this.bindEvents();
2038
2039 }
2040 };
2041
```

## language.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <translations>
3   <translation id="title">
4     <english>Home</english>
5     <italiano>Home</italiano>
6   </translation>
7   <translation id="btnAccount">
8     <english>Account</english>
9     <italiano>Profilo</italiano>
10  </translation>
11  <translation id="btnAddImmob">
12    <english>Inserting property</english>
13    <italiano>Inserimento immobile</italiano>
14  </translation>
15  <translation id="btnListaImmob">
16    <english>Properties saved</english>
17    <italiano>Immobili salvati</italiano>
18  </translation>
19  <translation id="btnInviaSchede">
20    <english>Send property</english>
21    <italiano>Invia immobili</italiano>
22  </translation>
23  <translation id="btnClearStorage">
24    <english>Clear Storage</english>
25    <italiano>Cancella memoria</italiano>
26  </translation>
27  <translation id="btnSearchDB">
28    <english>Query Server</english>
29    <italiano>Interroga Server</italiano>
30  </translation>
31  <translation id="connettendo">
32    <english>Realtor connecting to device</english>
33    <italiano>Realtor in connessione</italiano>
34  </translation>
35  <translation id="connesso">
36    <english>Realtor is Ready</english>
37    <italiano>Realtor è pronto</italiano>
38  </translation>
39  <translation id="PageAccount">
40    <english>Account</english>
41    <italiano>Profilo</italiano>
42  </translation>
43
```

## language.xml

```
44
45
46
47 <translation id="firstname">
48   <english>First name</english>
49   <italiano>Nome</italiano>
50 </translation>
51 <translation id="lastname">
52   <english>Last name</english>
53   <italiano>Cognome</italiano>
54 </translation>
55 <translation id="cpassword">
56   <english>Password confirm</english>
57   <italiano>Conferma password</italiano>
58 </translation>
59 <translation id="PageAccount">
60   <english>Account</english>
61   <italiano>Profilo</italiano>
62 </translation>
63 <translation id="code">
64   <english>Validation code</english>
65   <italiano>Codice di convalida</italiano>
66 </translation>
67
68
69
70
71 <translation id="btnInviaCodice">
72   <english>Post Code</english>
73   <italiano>Invia Codice</italiano>
74 </translation>
75 <translation id="btnConvalidaCodice">
76   <english>Validation</english>
77   <italiano>Convalida</italiano>
78 </translation>
79
80
81
82
83 <translation id="pageInsImmob">
84   <english>Inserting property</english>
85   <italiano>Inserimento immobile</italiano>
86 </translation>
```

## language.xml

```
87 <translation id="txtserachindirizzo">
88   <english>Address search</english>
89   <italiano>Ricerca indirizzo</italiano>
90 </translation>
91 <translation id="btnSalva">
92   <english>Save</english>
93   <italiano>Salva</italiano>
94 </translation>
95 <translation id="pageImmSalvati">
96   <english>Properties saved</english>
97   <italiano>Immobili salvati</italiano>
98 </translation>
99 <translation id="pagetabelle">
100  <english>List of Tables</english>
101  <italiano>Elenco Tabelle</italiano>
102 </translation>
103 <translation id="pageelencoSchedeServer">
104  <english>List of Cards</english>
105  <italiano>Elenco Schede</italiano>
106 </translation>
107
108
109 <translation id="txtcodename">
110  <english>Property Identification Code</english>
111  <italiano>Codice identificazione immobile</italiano>
112 </translation>
113 <translation id="txtcountry">
114  <english>Country</english>
115  <italiano>Stato</italiano>
116 </translation>
117 <translation id="txtadministrative_area_level_2">
118  <english>Second-order civil entity</english>
119  <italiano>Provincia</italiano>
120 </translation>
121 <translation id="txtadministrative_area_level_3">
122  <english>Third-order civil entity</english>
123  <italiano>Città</italiano>
124 </translation>
125 <translation id="txtlocality">
126  <english>Incorporated city</english>
127  <italiano>Comune</italiano>
128 </translation>
129 <translation id="txtroute">
```

## language.xml

```
130      <english>Route</english>
131      <italiano>Indirizzo</italiano>
132  </translation>
133  <translation id="txtstreet_number">
134      <english>Street number</english>
135      <italiano>Numero</italiano>
136  </translation>
137  <translation id="txtlatitude">
138      <english>Latitude</english>
139      <italiano>Latitudine</italiano>
140  </translation>
141  <translation id="txtlongitude">
142      <english>Longitude</english>
143      <italiano>Longitudine</italiano>
144  </translation>
145  <translation id="txtnote">
146      <english>Note</english>
147      <italiano>Note</italiano>
148  </translation>
149  <translation id="txtprice">
150      <english>Price</english>
151      <italiano>Prezzo</italiano>
152  </translation>
153  <translation id="txtprice">
154      <english>Price</english>
155      <italiano>Prezzo</italiano>
156  </translation>
157  <translation id="txtcurrency">
158      <english>Currency</english>
159      <italiano>Valuta</italiano>
160  </translation>
161
162  <translation id="txtbuilding">
163      <english>Land-use plans </english>
164      <italiano>Destinazione d'uso</italiano>
165  </translation>
166  <translation id="txtbuildingresidential">
167      <english>Building residential</english>
168      <italiano>Abitazione</italiano>
169  </translation>
170  <translation id="txtbuildingretail">
171      <english>Building retail</english>
172      <italiano>Locale commerciale</italiano>
```

## language.xml

```
173  </translation>
174  <translation id="txthouse">
175      <english>Construction type</english>
176      <italiano>Tipologia costruttiva</italiano>
177  </translation>
178  <translation id="txtapartmentbuilding">
179      <english>Apartment building</english>
180      <italiano>Condominio</italiano>
181  </translation>
182  <translation id="txttownhouse">
183      <english>Townhouse</english>
184      <italiano>Villa</italiano>
185  </translation>
186  <translation id="txtarea">
187      <english>Area</english>
188      <italiano>Zona</italiano>
189  </translation>
190  <translation id="txtsuburb">
191      <english>Suburb</english>
192      <italiano>Periferia</italiano>
193  </translation>
194  <translation id="txtcitycenter">
195      <english>City center</english>
196      <italiano>Centro</italiano>
197  </translation>
198  <translation id="txtsurface">
199      <english>Surface mq</english>
200      <italiano>Superficie mq</italiano>
201  </translation>
202  <translation id="txtrooms">
203      <english>Rooms</english>
204      <italiano>Stanze</italiano>
205  </translation>
206  <translation id="txtlevel">
207      <english>Level nr.</english>
208      <italiano>Piano</italiano>
209  </translation>
210  <translation id="txtelelevator">
211      <english>Elevator</english>
212      <italiano>Ascensore</italiano>
213  </translation>
214  <translation id="txtnotelevator">
215      <english>Not elevator</english>
```

## language.xml

```
216      <italiano>Senza ascensore</italiano>
217  </translation>
218  <translation id="txtbasement">
219      <english>Surface basement mq</english>
220      <italiano>Superficie cantina mq</italiano>
221  </translation>
222  <translation id="txtcarpark">
223      <english>Surface car park mq</english>
224      <italiano>Superficie parcheggio mq</italiano>
225  </translation>
226  <translation id="txtbox">
227      <english>Surface box mq</english>
228      <italiano>Superficie box mq</italiano>
229  </translation>
230  <translation id="txtsurfaceinteriorbalconies">
231      <english>Surface interior balconies mq</english>
232      <italiano>Superficie balconi interni mq</italiano>
233  </translation>
234  <translation id="txtsurfaceexteriorbalconies">
235      <english>Surface exterior balconies mq</english>
236      <italiano>Superficie balconi esterni mq</italiano>
237  </translation>
238  <translation id="txtgardenapartment">
239      <english>Surface garden apartment mq</english>
240      <italiano>Superficie giardino privato mq</italiano>
241  </translation>
242  <translation id="txtnosharedgarden">
243      <english>No shared garden</english>
244      <italiano>Assenza giardino condominiale</italiano>
245  </translation>
246  <translation id="txtsharedgarden">
247      <english>Shared garden</english>
248      <italiano>Giardino condominiale</italiano>
249  </translation>
250  <translation id="txtnowaterandsanitarysystem">
251      <english>No water and sanitary system</english>
252      <italiano>Assenza impianto idrico-sanitario</italiano>
253  </translation>
254  <translation id="txtwaterandsanitarysystem">
255      <english>Water and sanitary system</english>
256      <italiano>Impianto idrico-sanitario</italiano>
257  </translation>
258  <translation id="txtnowatertank">
```

language.xml

```
259      <english>No water tank</english>
260      <italiano>Assenza riserva idrica</italiano>
261  </translation>
262  <translation id="txtwatertank">
263      <english>Water tank</english>
264      <italiano>Riserva idrica</italiano>
265  </translation>
266  <translation id="txtsewerage">
267      <english>Sewerage</english>
268      <italiano>Fogna pubblica</italiano>
269  </translation>
270  <translation id="txtcesspool">
271      <english>Cesspool</english>
272      <italiano>Pozzo nero</italiano>
273  </translation>
274  <translation id="txtnosewerage">
275      <english>No sewerage</english>
276      <italiano>Assenza fogna</italiano>
277  </translation>
278  <translation id="txtnumberwell">
279      <english>Number well</english>
280      <italiano>Numero pozzi artesiani</italiano>
281  </translation>
282  <translation id="txtelelectricalsystem">
283      <english>electrical system</english>
284      <italiano>Impianto elettrico</italiano>
285  </translation>
286  <translation id="txtnoelectricalsystem">
287      <english>No electrical system</english>
288      <italiano>Assenza impianto elettrico</italiano>
289  </translation>
290  <translation id="txtgasline">
291      <english>Gas line</english>
292      <italiano>Impianto gas</italiano>
293  </translation>
294  <translation id="txtnogasline">
295      <english>No gas line</english>
296      <italiano>Assenza impianto gas</italiano>
297  </translation>
298  <translation id="txtintercomsystem">
299      <english>Intercom system</english>
300      <italiano>Impianto citofonico </italiano>
301  </translation>
```

## language.xml

```
302 <translation id="txtnointercomsystem">
303   <english>No intercom system</english>
304   <italiano>Assenza impianto citofonico </italiano>
305 </translation>
306 <translation id="txtallarmsystem">
307   <english>Allarm system</english>
308   <italiano>Impianto allarme</italiano>
309 </translation>
310 <translation id="txtnoallarmsystem">
311   <english>No allarm system</english>
312   <italiano>Assenza impianto allarme</italiano>
313 </translation>
314 <translation id="txtindependentheatingsystem">
315   <english>Independent heating system</english>
316   <italiano>Impianto di riscaldamento autonomo</italiano>
317 </translation>
318 <translation id="txtcentralheatingsystem">
319   <english>Central heating system</english>
320   <italiano>Impianto di riscaldamento centralizzato</italiano>
321 </translation>
322 <translation id="txtnoheatingsystem">
323   <english>No heating system</english>
324   <italiano>Assenza impianto di riscaldamento</italiano>
325 </translation>
326 <translation id="txtcoolingsystem">
327   <english>Cooling system </english>
328   <italiano>Impianto di raffrescamento</italiano>
329 </translation>
330 <translation id="txtnocoolingsystem">
331   <english>No cooling system </english>
332   <italiano>Assenza impianto di raffrescamento</italiano>
333 </translation>
334 <translation id="txtgoodstate">
335   <english>Apartment in a good state of repair </english>
336   <italiano>Appartamento in ottimo stato</italiano>
337 </translation>
338 <translation id="txtreasonableestate">
339   <english>Apartment in a reasonable state of repair </english>
340   <italiano>Appartamento in discreto stato</italiano>
341 </translation>
342 <translation id="txtbadstate">
343   <english>Apartment in a bad state of repair </english>
344   <italiano>Appartamento in pessimo stato</italiano>
```

## language.xml

```
345  </translation>
346  <translation id="txtgoodstatehome">
347      <english>Home in a good state of repair </english>
348      <italiano>Edificio in ottimo stato</italiano>
349  </translation>
350  <translation id="txtreasonablestatehome">
351      <english>Home in a reasonable state of repair </english>
352      <italiano>Edificio in discreto stato</italiano>
353  </translation>
354  <translation id="txtbadstatehome">
355      <english>Home in a bad state of repair </english>
356      <italiano>Edificio in pessimo stato</italiano>
357  </translation>
358  <translation id="txtnoancientbuilding">
359      <english>Not ancient building </english>
360      <italiano>Edificio non di pregio storico</italiano>
361  </translation>
362  <translation id="txtancientbuilding">
363      <english>Ancient building </english>
364      <italiano>Edificio di pregio storico</italiano>
365  </translation>
366  <translation id="txtnolandscape">
367      <english>Not landscape</english>
368      <italiano>Nessun paesaggio</italiano>
369  </translation>
370  <translation id="txtlandscape">
371      <english>Landscape</english>
372      <italiano>Paesaggio</italiano>
373  </translation>
374  <translation id="txtnostreetlighting">
375      <english>Not street lighting</english>
376      <italiano>Assenza illuminazione pubblica </italiano>
377  </translation>
378  <translation id="txtstreetlighting">
379      <english>Street lighting</english>
380      <italiano>Illuminazione pubblica </italiano>
381  </translation>
382  <translation id="txtasphaltroad">
383      <english>Asphalt road</english>
384      <italiano>Strada asfaltata</italiano>
385  </translation>
386  <translation id="txtdirtroad">
387      <english>Dirt road</english>
```

language.xml

```
388      <italiano>Strada sterrata</italiano>
389  </translation>
390  <translation id="txtservices">
391      <english>Services</english>
392      <italiano>Servizi</italiano>
393  </translation>
394  <translation id="txtyes">
395      <english>Yes</english>
396      <italiano>Si</italiano>
397  </translation>
398  <translation id="txtno">
399      <english>No</english>
400      <italiano>No</italiano>
401  </translation>
402  <translation id="txtprimaryschool">
403      <english>Primary school</english>
404      <italiano>Scuola primaria</italiano>
405  </translation>
406  <translation id="txtsecondaryschool">
407      <english>Secondary school</english>
408      <italiano>Scuola secondaria</italiano>
409  </translation>
410  <translation id="txtculturalcentre">
411      <english>Cultural centre</english>
412      <italiano>Centro culturale</italiano>
413  </translation>
414  <translation id="txthospital">
415      <english>Hospital</english>
416      <italiano>Ospedale </italiano>
417  </translation>
418  <translation id="txtbus">
419      <english>Bus</english>
420      <italiano>Autobus</italiano>
421  </translation>
422  <translation id="txtrailway">
423      <english>Railway</english>
424      <italiano>Ferrovia</italiano>
425  </translation>
426  <translation id="txtairport">
427      <english>Airport</english>
428      <italiano>Aeroporto</italiano>
429  </translation>
430  <translation id="txtsubway">
```

## language.xml

```
431      <english>Subway</english>
432      <italiano>Metropolitana</italiano>
433  </translation>
434
435
436
437
438
439  <translation label="scheda">
440      <english>Cards</english>
441      <italiano>Schede</italiano>
442  </translation>
443  <translation label="tabella">
444      <english>Table</english>
445      <italiano>Tabella</italiano>
446  </translation>
447  <translation label="Reset totale?">
448      <english>Total reset?</english>
449      <italiano>Reset totale?</italiano>
450  </translation>
451 </translation>
452  <translation label="Codice di convalida account per l'app Realtor">
453      <english>Validation Code account for the app Realtor</english>
454      <italiano>Codice di convalida account per l'app Realtor</italiano>
455  </translation>
456 </translation>
457  <translation label="Email inviated">
458      <english>Email inviated</english>
459      <italiano>"Email inviata"</italiano>
460  </translation>
461 </translation>
462  <translation label="Email not inviated">
463      <english>Email not inviated"</english>
464      <italiano>Email non inviata</italiano>
465  </translation>
466 </translation>
467  <translation label="Password non corrispondono!">
468      <english>Passwords do not match!</english>
469      <italiano>Password non corrispondono!</italiano>
470  </translation>
471 </translation>
472  <translation label="Email non valida!">
473      <english>Invalid Email!</english>
```

## language.xml

```
474      <italiano>Email non valida!</italiano>
475  </translation>
476 </translation>
477  <translation label="Connessione Internet non disponibile!">
478      <english>Wireless internet is available!</english>
479      <italiano>Connessione Internet non disponibile!</italiano>
480  </translation>
481 </translation>
482  <translation label="Tutti i campi devono essere pieni!">
483      <english>All fields must be filled!</english>
484      <italiano>Tutti i campi devono essere pieni!</italiano>
485  </translation>
486 </translation>
487  <translation label="Nickname non valido">
488      <english>Nickname already exists or invalid!</english>
489      <italiano>Nickname già esistente o non valido!</italiano>
490  </translation>
491 </translation>
492  <translation label="Nickname già esiste">
493      <english>Nickname already exists!</english>
494      <italiano>Nickname già esistente!</italiano>
495  </translation>
496 </translation>
497  <translation label="Codice valido">
498      <english>Code is valid!</english>
499      <italiano>Codice valido!</italiano>
500  </translation>
501 </translation>
502  <translation label="Codice non valido">
503      <english>Code is not valid!</english>
504      <italiano>Codice non valido!</italiano>
505  </translation>
506 </translation>
507  <translation label="Account creato">
508      <english>Account created!</english>
509      <italiano>Profilo creato!</italiano>
510  </translation>
511 </translation>
512  <translation label="Account non creato">
513      <english>Account not created!</english>
514      <italiano>Profilo non creato!</italiano>
515  </translation>
516 </translation>
```

## language.xml

```
517 <translation label="Salvataggio effettuato!">
518     <english>Save made!</english>
519     <italiano>Salvataggio effettuato!</italiano>
520 </translation>
521 </translation>
522 <translation label="Confermi l'eliminazione della scheda?">
523     <english>Confirm deletion of the card?</english>
524     <italiano>Confermi l'eliminazione della scheda?</italiano>
525 </translation>
526 </translation>
527 <translation label="Confermi l'invio delle schede?">
528     <english>Do confirm the sending of cards?</english>
529     <italiano>Confermi l'invio delle schede?</italiano>
530 </translation>
531 </translation>
532 <translation label="Nessuna scheda memorizzata">
533     <english>No entries stored</english>
534     <italiano>Nessuna scheda memorizzata</italiano>
535 </translation>
536 <translation label="Nessuna scheda selezionata">
537     <english>No selected tab</english>
538     <italiano>Nessuna scheda selezionata</italiano>
539 </translation>
540 </translation>
541 <translation label="Vuoi uscire dall'applicazione?">
542     <english>Do you want to exit the application?</english>
543     <italiano>Vuoi uscire dall'applicazione?</italiano>
544 </translation>
545 </translation>
546 <translation label="posizione non autorizzata">
547     <english>Acquiring the current location is not authorized</english>
548     <italiano>Acquisizione della posizione corrente non autorizzata
549     </italiano>
550 </translation>
551 <translation label="posizione non disponibile">
552     <english>Is not available the detection of the current position
553     </english>
554     <italiano>Non è disponibile la rilevazione della posizione corrente
555     </italiano>
556 </translation>
557 <translation label="timeout posizione">
558     <english>Timeout detecting current position</english>
559     <italiano>Timeout rilevazione posizione corrente</italiano>
```

## language.xml

```
560 </translation>
561 <translation label="Manca codice immobile">
562   <english>Missing building code</english>
563   <italiano>Manca codice immobile</italiano>
564 </translation>
565 <translation label="Recensione per id">
566   <english>Review for id:</english>
567   <italiano>Recensione per id: </italiano>
568 </translation>
569 <translation label="Sono una nuova recensione?">
570   <english>I am a new review?</english>
571   <italiano>Sono una nuova recensione?</italiano>
572 </translation>
573 <translation label="Vuoi modificare questa scheda?">
574   <english>Want to edit this card?</english>
575   <italiano>Vuoi modificare questa scheda?</italiano>
576 </translation>
577
578 </translation>
579 <translation label="Richiesta contatto per informazione su Realtor">
580   <english>
581     Request for contact information on Realtor
582   <english>
583     <italiano>Richiesta contatto per informazione su Realtor</italiano>
584 </translation>
585 </translation>
586 <translation label="Richiesta contatto per informazione su questa scheda">
587   <english>Request for contact information about this card</english>
588   <italiano>Richiesta contatto per informazione su questa scheda
589   </italiano>
590 </translation>
591
592
593
594 </translations>
```

## index.php

```
1 <?php
2
3 require 'Slim/Slim.php';
4
5 $app = new Slim();
6
7 $app->post('/scheda', 'addscheda');
8 $app->post('/account', 'addaccount');
9 $app->get('/account/:nickname/:emailuser', 'checknick');
10 $app->post('/mailto', 'mailto');
11 $app->post('/testjson', 'testjson');
12 $app->get('/showdb', 'showdb');
13 $app->post('/showschede', 'showschede');
14 $app->get('/scheda/:table/:id', 'scheda');
15
16 $app->run();
17
18
19 function testjson() {
20 /**
21 *      solo per testare curl
22 */
23
24 $request = Slim::getInstance()->request();
25 echo 'Slim: ' . json_encode($request);
26 $jsonondec = json_decode($request->getBody());
27 echo 'json: ' . json_encode($jsonondec);
28
29 }
30
31 function showdb() {
32 /**
33 * mostro tutte le tabelle
34 */
35
36 try {
37     $sql = "SHOW TABLES LIKE 'DB_%'";
38     $db = getConnection();
39     $stmt = $db->prepare($sql);
40     $stmt->execute();
41     $result = $stmt->fetchALL(PDO::FETCH_NUM);
42     $db = null;
43     echo(json_encode($result));
```

## index.php

```
44
45
46
47 } catch(PDOException $e) {
48
49     echo '{"error":{"text":'. $e->getMessage() .'}}';
50 }
51 }
52
53 function scheda($table,$id) {
54 /**
55 * mostro tutte la scheda con id nel db
56 */
57
58 try {
59     $sql = "SELECT * FROM ".$table." WHERE ID=:id";
60     //echo $sql;
61     $db = getConnection();
62     $stmt = $db->prepare($sql);
63     $stmt->bindParam("id", $id);
64     $stmt->execute();
65     $result = $stmt->fetchObject();
66     $db = null;
67     echo(json_encode($result));
68
69 } catch(PDOException $e) {
70
71     echo '{"error":{"text":'. $e->getMessage() .'}}';
72 }
73 }
74
75
76 function showschede() {
77 /**
78 * mostro tutte le schede del db
79 */
80 $request = Slim::getInstance()->request();
81 $input = json_decode($request->getBody());
82 $table = $input -> nome;
83 $lowlimit = $input -> lowlimit;
84 $highlimit = $input -> highlimit;
85 $coord[0] = $input -> coord[0];
86 $coord[1] = $input -> coord[1];
```

## index.php

```
87 $coord[2] = $input -> coord[2];
88 $coord[3] = $input -> coord[3];
89
90
91 try {
92     $sql = "SELECT
93         ID,administrative_area_level_3,locality,route,street_number,longitude,latitude
94         FROM ".$table."
95         WHERE ((longitude BETWEEN ".$coord[0].") AND ".$coord[1].")AND(latitude BETWEEN
96         ".$coord[2].") AND ".$coord[3]." )) ORDER BY date DESC LIMIT ".$lowlimit.", ".
97         $highlimit;
98
99     //echo $sql;
100    $db = getConnection();
101    $stmt = $db->prepare($sql);
102    $stmt->execute();
103    $result = $stmt->fetchALL(PDO::FETCH_NUM);
104    $db = null;
105    echo(json_encode($result));
106
107 } catch(PDOException $e) {
108
109     echo '{"error":{"text":'. $e->getMessage() .'}}';
110 }
111
112 try {
113     $db = getConnection();
114
115     $sql = "CREATE TABLE IF NOT EXISTS usertable (
116         id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
117         nickname VARCHAR(50) NOT NULL,
118         firstname VARCHAR(50) NOT NULL,
119         lastname VARCHAR(50) NOT NULL,
120         emailuser VARCHAR(50) NOT NULL,
121         password VARCHAR(50) NOT NULL
122         ) CHARACTER SET utf8 COLLATE utf8_general_ci";
123     $stmt = $db->prepare($sql);
124     $stmt->execute();
125 }
```

## index.php

```
126     $sql = "SELECT * FROM usertable WHERE nickname = :nickname and emailuser
127         <>> :emailuser";
128     $stmt = $db->prepare($sql);
129     $stmt->bindParam("nickname", $nickname);
130     $stmt->bindParam("emailuser", $emailuser);
131     $stmt->execute();
132     $result = $stmt->fetchAll(PDO::FETCH_OBJ);
133     //echo json_encode($result);
134     $db = null;
135     if ($result == null){
136         echo "false";
137     }else{
138         echo "true";
139     }
140 } catch(PDOException $e) {
141
142     echo '{"error":{"text":' . $e->getMessage() . '}}';
143 }
144 }
145
146 function addaccount() {
147 /**
148 * Se sono arrivato ho superato il check email allora
149 * aggiorno o aggiungo l'utente
150 */
151
152 $request = Slim::getInstance()->request();
153 $adduser = json_decode($request->getBody());
154 //echo json_encode($adduser);
155 try {
156     $db = getConnection();
157
158     $sql = "CREATE TABLE IF NOT EXISTS usertable (
159         id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
160         nickname VARCHAR(50) NOT NULL,
161         firstname VARCHAR(50) NOT NULL,
162         lastname VARCHAR(50) NOT NULL,
163         emailuser VARCHAR(50) NOT NULL,
164         password VARCHAR(50) NOT NULL
165     ) CHARACTER SET utf8 COLLATE utf8_general_ci";
166     $stmt = $db->prepare($sql);
167     $stmt->execute();
```

## index.php

```
168
169     $sql = "SELECT * FROM usertable WHERE emailuser = :emailuser";
170     $stmt = $db->prepare($sql);
171     $stmt->bindParam("emailuser", $adduser->emailuser);
172     $stmt->execute();
173     $result = $stmt->fetchObject();
174     //echo json_encode($result);
175     // $db = null; //inutile
176     if ($result == null){
177         // echo "email utente non esiste";
178         $sql = "INSERT INTO usertable (nickname, firstname, lastname, emailuser, password)
179             VALUES (:nickname, :firstname, :lastname, :emailuser, :password)";
180     } else{
181         // echo "email utente esiste";
182
183         $sql = "UPDATE usertable SET nickname=:nickname, firstname=:firstname,
184             lastname=:lastname, password=:password WHERE emailuser=:emailuser";
185
186     }
187
188
189     // $db = getConnection(); //inutile
190     $stmt = $db->prepare($sql);
191     $stmt->bindParam("nickname", $adduser->nickname);
192     $stmt->bindParam("firstname", $adduser->firstname);
193     $stmt->bindParam("lastname", $adduser->lastname);
194     $stmt->bindParam("emailuser", $adduser->emailuser);
195     $stmt->bindParam("password", $adduser->password);
196
197     $stmt->execute();
198     $result->id = $db->lastInsertId();
199
200     $db = null;
201     echo json_encode($result);
202
203
204 } catch(PDOException $e) {
205
206     echo '{"error":{"text":' . $e->getMessage() . '}}';
207 }
208 }
```

## index.php

```
209
210
211 function addscheda() {
212
213
214     $request = Slim::getInstance()->request();
215     $scheda = json_decode($request->getBody());
216
217     try {
218         $db = getConnection();
219
220
221
222         $table = strtoupper("DB_".($scheda->country)."_" .($scheda-
223             >administrative_area_level_2));
224
225         $sql = "CREATE TABLE IF NOT EXISTS " . $table . " (
226             ID BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
227             link BIGINT NULL,
228             date DATE NOT NULL,
229             nickname VARCHAR(50) NOT NULL,
230             codename VARCHAR(50) NOT NULL,
231             longitude FLOAT NOT NULL,
232             latitude FLOAT NOT NULL,
233             country VARCHAR(50) NULL,
234             administrative_area_level_2 VARCHAR(50) NULL,
235             administrative_area_level_3 VARCHAR(50) NULL,
236             locality VARCHAR(50) NULL,
237             route VARCHAR(50) NULL,
238             street_number VARCHAR(50) NULL,
239             buildingretail TINYINT NULL,
240             townhouse TINYINT NULL,
241             citycenter TINYINT NULL,
242             surface MEDIUMINT NULL,
243             rooms TINYINT NULL,
244             level TINYINT NULL,
245             elevator TINYINT NULL,
246             basement MEDIUMINT NULL,
247             carpark MEDIUMINT NULL,
248             box MEDIUMINT NULL,
249             surfaceinteriorbalconies MEDIUMINT NULL,
250             surfaceexteriorbalconies MEDIUMINT NULL,
```

## index.php

```
251     gardenapartment INT NULL,  
252     sharedgarden TINYINT NULL,  
253     waterandsanitarysystem TINYINT NULL,  
254     watertank TINYINT NULL,  
255     sewerage TINYINT NULL,  
256     numberwell TINYINT NULL,  
257     electricalsystem TINYINT NULL,  
258     gasline TINYINT NULL,  
259     intercomsystem TINYINT NULL,  
260     allarmsystem TINYINT NULL,  
261     heatingsystem TINYINT NULL,  
262     coolingsystem TINYINT NULL,  
263     apartmentsstateofrepair TINYINT NULL,  
264     homestateofrepair TINYINT NULL,  
265     ancientbuilding TINYINT NULL,  
266     landscape TINYINT NULL,  
267     streetlighting TINYINT NULL,  
268     roads TINYINT NULL,  
269     primarieschool TINYINT NULL,  
270     secondarieschool TINYINT NULL,  
271     culturalcentre TINYINT NULL,  
272     hospital TINYINT NULL,  
273     bus TINYINT NULL,  
274     railway TINYINT NULL,  
275     airport TINYINT NULL,  
276     subway TINYINT NULL,  
277     note VARCHAR(100) NULL,  
278     price FLOAT NOT NULL,  
279     currency VARCHAR(10) NULL,  
280     normalized INT NULL,  
281     photo_1 MEDIUMBLOB NULL,  
282     photo_2 MEDIUMBLOB NULL,  
283     photo_3 MEDIUMBLOB NULL,  
284     photo_4 MEDIUMBLOB NULL,  
285     photo_5 MEDIUMBLOB NULL,  
286     photo_6 MEDIUMBLOB NULL,  
287     photo_7 MEDIUMBLOB NULL,  
288     photo_8 MEDIUMBLOB NULL,  
289     photo_9 MEDIUMBLOB NULL,  
290     photo_10 MEDIUMBLOB NULL  
291     ) CHARACTER SET utf8 COLLATE utf8_general_ci";  
292     $stmt = $db->prepare($sql);  
293     $stmt->execute();
```

## index.php

```
294
295
296
297     $sql = "SELECT * FROM " . $table . " WHERE nickname = :nickname AND ID = :ID
298     AND link = :link";
299     $stmt = $db->prepare($sql);
300     $stmt->bindParam("nickname", $scheda->nickname);
301     $stmt->bindParam("ID", $scheda->ID);
302     $stmt->bindParam("link", $scheda->link);
303     $stmt->execute();
304     $result = $stmt->fetchObject();
305     //echo json_encode($result);
306     if ($result == null){
307         //echo "INSERISCO NUOVO ELEMENTO";
308         $sql = "INSERT INTO ".$table." (
309             link,
310             date,
311             nickname,
312             codename,
313             longitude,
314             latitude,
315             country,
316             administrative_area_level_2,
317             administrative_area_level_3,
318             locality,
319             route,
320             street_number,
321             buildingretail,
322             townhouse,
323             citycenter,
324             surface,
325             rooms,
326             level,
327             elevator,
328             basement,
329             carpark,
330             box,
331             surfaceinteriorbalconies,
332             surfaceexteriorbalconies,
333             gardenapartment,
334             sharedgarden,
335             waterandsanitarysystem,
336             watertank,
```

## index.php

```
336     sewerage,  
337     numberwell,  
338     electricalsystem,  
339     gasline,  
340     intercomsystem,  
341     allarmsystem,  
342     heatingsystem,  
343     coolingsystem,  
344     apartmentstateofrepair,  
345     homestateofrepair,  
346     ancientbuilding,  
347     landscape,  
348     streetlighting,  
349     roads,  
350     primaryschool,  
351     secondaryschool,  
352     culturalcentre,  
353     hospital,  
354     bus,  
355     railway,  
356     airport,  
357     subway,  
358     note,  
359     price,  
360     currency,  
361     normalized,  
362     photo_1,  
363     photo_2,  
364     photo_3,  
365     photo_4,  
366     photo_5,  
367     photo_6,  
368     photo_7,  
369     photo_8,  
370     photo_9,  
371     photo_10  
372     ) VALUES (  
373     :link,  
374     CURDATE(),  
375     :nickname,  
376     :codename,  
377     :longitude,  
378     :latitude,
```

## index.php

```
379 :country,  
380 :administrative_area_level_2,  
381 :administrative_area_level_3,  
382 :locality,  
383 :route,  
384 :street_number,  
385 :buildingretail,  
386 :townhouse,  
387 :citycenter,  
388 :surface,  
389 :rooms,  
390 :level,  
391 :elevator,  
392 :basement,  
393 :carpark,  
394 :box,  
395 :surfaceinteriorbalconies,  
396 :surfaceexteriorbalconies,  
397 :gardenapartment,  
398 :sharedgarden,  
399 :waterandsanitarysystem,  
400 :watertank,  
401 :sewerage,  
402 :numberwell,  
403 :electricalsystem,  
404 :gasline,  
405 :intercomsystem,  
406 :allarmsystem,  
407 :heatingsystem,  
408 :coolingsystem,  
409 :apartmentstateofrepair,  
410 :homestateofrepair,  
411 :ancientbuilding,  
412 :landscape,  
413 :streetlighting,  
414 :roads,  
415 :primaryschool,  
416 :secondaryschool,  
417 :culturalcentre,  
418 :hospital,  
419 :bus,  
420 :railway,  
421 :airport,
```

## index.php

```
422    :subway,  
423    :note,  
424    :price,  
425    :currency,  
426    :normalized,  
427    :photo_1,  
428    :photo_2,  
429    :photo_3,  
430    :photo_4,  
431    :photo_5,  
432    :photo_6,  
433    :photo_7,  
434    :photo_8,  
435    :photo_9,  
436    :photo_10  
437    );  
438  
439 }else{  
440     //echo ("AGGIORNO ELEMENTO: " . $result->ID);  
441  
442     $sql = "UPDATE ".$table." SET  
443         link=:link,  
444         date=CURDATE(),  
445         nickname=:nickname,  
446         codename=:codename,  
447         currency=:currency,  
448         longitude=:longitude,  
449         latitude=:latitude,  
450         country=:country,  
451         administrative_area_level_2=:administrative_area_level_2,  
452         administrative_area_level_3=:administrative_area_level_3,  
453         locality=:locality,  
454         route=:route,  
455         street_number=:street_number,  
456         buildingretail=:buildingretail,  
457         townhouse=:townhouse,  
458         citycenter=:citycenter,  
459         surface=:surface,  
460         rooms=:rooms,  
461         level=:level,  
462         elevator=:elevator,  
463         basement=:basement,  
464         carpark=:carpark,
```

index.php

```
465    box=:box,
466    surfaceinteriorbalconies=:surfaceinteriorbalconies,
467    surfaceexteriorbalconies=:surfaceexteriorbalconies,
468    gardenapartment=:gardenapartment,
469    sharedgarden=:sharedgarden,
470    waterandsanitarysystem=:waterandsanitarysystem,
471    watertank=:watertank,
472    sewerage=:sewerage,
473    numberwell=:numberwell,
474    electricalsystem=:electricalsystem,
475    gasline=:gasline,
476    intercomsystem=:intercomsystem,
477    allarmsystem=:allarmsystem,
478    heatingsystem=:heatingsystem,
479    coolingsystem=:coolingsystem,
480    apartmentstateofrepair=:apartmentstateofrepair,
481    homestateofrepair=:homestateofrepair,
482    ancientbuilding=:ancientbuilding,
483    landscape=:landscape,
484    streetlighting=:streetlighting,
485    roads=:roads,
486    primaryschool=:primaryschool,
487    secondaryschool=:secondaryschool,
488    culturalcentre=:culturalcentre,
489    hospital=:hospital,
490    bus=:bus,
491    railway=:railway,
492    airport=:airport,
493    subway=:subway,
494    note=:note,
495    price=:price,
496    normalized=:normalized,
497        photo_1=:photo_1,
498        photo_2=:photo_2,
499        photo_3=:photo_3,
500        photo_4=:photo_4,
501        photo_5=:photo_5,
502        photo_6=:photo_6,
503        photo_7=:photo_7,
504        photo_8=:photo_8,
505        photo_9=:photo_9,
506        photo_10=:photo_10
507    WHERE ID=". $result->ID;
```

index.php

```
508      }
509 if( ($scheda-> surface)>0 ) {
510
511 $scheda->normalized = $scheda-> surface;
512
513 $scheda->normalized = ($scheda->normalized) + ($scheda-> basement)*0.3;
514 $scheda->normalized = ($scheda->normalized) + ($scheda-> carpark)*0.3;
515 $scheda->normalized = ($scheda->normalized) + ($scheda-> box)*0.5;
516 $scheda->normalized = ($scheda->normalized) + ($scheda-> surfaceinteriorbalconies)*0.1;
517 $scheda->normalized = ($scheda->normalized) + ($scheda-> surfaceexteriorbalconies)*0.2;
518 $scheda->normalized = ($scheda->normalized) + ($scheda-> gardenapartment)*0.2;
519
520
521 $k = (($scheda-> buildingretail) * 1) +
522   (($scheda-> level) * ($scheda-> elevator) * 1) +
523   (($scheda-> townhouse) * 1) +
524   (($scheda-> citycenter) * 1) +
525   (($scheda-> sharedgarden) * 1) +
526   (($scheda-> waterandsanitarysystem) * 1) +
527   (($scheda-> watertank) * 1) +
528   (($scheda-> sewerage) * 1) +
529   (($scheda-> numberwell) * 1) +
530   (($scheda-> electricalsystem) * 1) +
531   (($scheda-> gasline) * 1) +
532   (($scheda-> intercomsystem) * 1) +
533   (($scheda-> allarmsystem) * 1) +
534   (($scheda-> heatingsystem) * 1) +
535   (($scheda-> coolingsystem) * 1) +
536   (($scheda-> apartmentstateofrepair) * 1) +
537   (($scheda-> homestateofrepair) * 1) +
538   (($scheda-> ancientbuilding) * 1) +
539   (($scheda-> landscape) * 1) +
540   (($scheda-> streetlighting) * 1) +
541   (($scheda-> roads) * 1) +
542   (($scheda-> primaryschool) * 1) +
543   (($scheda-> secondaryschool) * 1) +
544   (($scheda-> culturalcentre) * 1) +
545   (($scheda-> hospital) * 1) +
546   (($scheda-> bus) * 1) +
547   (($scheda-> railway) * 1) +
548   (($scheda-> airport) * 1) +
549   (($scheda-> subway) * 1);
550
```

## index.php

```
551
552 //echo $scheda->normalized;
553 //echo "\n";
554
555 $scheda->normalized = ($scheda->normalized) *(1+$k/30);
556
557 //echo $k;
558 //echo "\n";
559 //echo $scheda->normalized;
560
561
562 }
563     // $sql = "INSERT INTO ".$table." (date, nickname, codename, note, price, currency,
      longitude, latitude, administrative_area_level_3, locality, route,
      street_number, photo_1, photo_2, photo_3, photo_4, photo_5, photo_6, photo_7, photo_8, photo_9,
      photo_10) VALUES
      (CURDATE(), :nickname, :codename, :note, :price, :currency, :longitude, :latitude, :administrati
      ve_area_level_3, :locality, :route, :street_number, :photo_1, :photo_2, :photo_3, :photo_4, :phot
      o_5, :photo_6, :photo_7, :photo_8, :photo_9, :photo_10)";
564     //echo $sql;
565     //echo "\n";
566     $stmt = $db->prepare($sql);
567
568     $stmt->bindParam("link", $scheda->link);
569
570     $stmt->bindParam("nickname", $scheda->nickname);
571     $stmt->bindParam("codename", $scheda->codename);
572     $stmt->bindParam("longitude", $scheda->longitude);
573     $stmt->bindParam("latitude", $scheda-> latitude);
574     $stmt->bindParam("country", $scheda-> country);
575     $stmt->bindParam("administrative_area_level_2", $scheda->
      administrative_area_level_2);
576     $stmt->bindParam("administrative_area_level_3", $scheda->
      administrative_area_level_3);
577     $stmt->bindParam("locality", $scheda-> locality);
578     $stmt->bindParam("route", $scheda-> route);
579     $stmt->bindParam("street_number", $scheda-> street_number);
580
581     $stmt->bindParam("buildingretail", $scheda-> buildingretail);
582     $stmt->bindParam("townhouse", $scheda-> townhouse);
583     $stmt->bindParam("citycenter", $scheda-> citycenter);
584     $stmt->bindParam("surface", $scheda-> surface);
585     $stmt->bindParam("rooms", $scheda-> rooms);
```

## index.php

```
586     $stmt->bindParam("level", $scheda-> level);
587     $stmt->bindParam("elevator", $scheda-> elevator);
588     $stmt->bindParam("basement", $scheda-> basement);
589     $stmt->bindParam("carpark", $scheda-> carpark);
590     $stmt->bindParam("box", $scheda-> box);
591     $stmt->bindParam("surfaceinteriorbalconies", $scheda-> surfaceinteriorbalconies);
592     $stmt->bindParam("surfaceexteriorbalconies", $scheda-> surfaceexteriorbalconies);
593     $stmt->bindParam("gardenapartment", $scheda-> gardenapartment);
594     $stmt->bindParam("sharedgarden", $scheda-> sharedgarden);
595     $stmt->bindParam("waterandsanitarysystem", $scheda-> waterandsanitarysystem);
596     $stmt->bindParam("watertank", $scheda-> watertank);
597     $stmt->bindParam("sewerage", $scheda-> sewerage);
598     $stmt->bindParam("numberwell", $scheda-> numberwell);
599     $stmt->bindParam("electricalsystem", $scheda-> electricalsystem);
600     $stmt->bindParam("gasline", $scheda-> gasline);
601     $stmt->bindParam("intercomsystem", $scheda-> intercomsystem);
602     $stmt->bindParam("allarmsystem", $scheda-> allarmsystem);
603     $stmt->bindParam("heatingsystem", $scheda-> heatingsystem);
604     $stmt->bindParam("coolingsystem", $scheda-> coolingsystem);
605     $stmt->bindParam("apartmentstateofrepair", $scheda-> apartmentstateofrepair);
606     $stmt->bindParam("homestateofrepair", $scheda-> homestateofrepair);
607     $stmt->bindParam("ancientbuilding", $scheda-> ancientbuilding);
608     $stmt->bindParam("landscape", $scheda-> landscape);
609     $stmt->bindParam("streetlighting", $scheda-> streetlighting);
610     $stmt->bindParam("roads", $scheda-> roads);
611     $stmt->bindParam("primaryschool", $scheda-> primaryschool);
612     $stmt->bindParam("secondaryschool", $scheda-> secondaryschool);
613     $stmt->bindParam("culturalcentre", $scheda-> culturalcentre);
614     $stmt->bindParam("hospital", $scheda-> hospital);
615     $stmt->bindParam("bus", $scheda-> bus);
616     $stmt->bindParam("railway", $scheda-> railway);
617     $stmt->bindParam("airport", $scheda-> airport);
618     $stmt->bindParam("subway", $scheda-> subway);
619
620     $stmt->bindParam("note", $scheda->note);
621     $stmt->bindParam("price", $scheda->price);
622     $stmt->bindParam("currency", $scheda->currency);
623     $stmt->bindParam("normalized", $scheda->normalized);
624     //echo json_encode($scheda-> photo);
625
626     if (($scheda-> photo) == null){
627         //echo "non ci sono foto";
628         $stmt->bindValue("photo_1", null);
```

## index.php

```
629     $stmt->bindValue("photo_2", null);
630     $stmt->bindValue("photo_3", null);
631     $stmt->bindValue("photo_4", null);
632     $stmt->bindValue("photo_5", null);
633     $stmt->bindValue("photo_6", null);
634     $stmt->bindValue("photo_7", null);
635     $stmt->bindValue("photo_8", null);
636     $stmt->bindValue("photo_9", null);
637     $stmt->bindValue("photo_10", null);
638
639 } else {
640     $stmt->bindParam("photo_1", $scheda-> photo[0]);
641     $stmt->bindParam("photo_2", $scheda-> photo[1]);
642     $stmt->bindParam("photo_3", $scheda-> photo[2]);
643     $stmt->bindParam("photo_4", $scheda-> photo[3]);
644     $stmt->bindParam("photo_5", $scheda-> photo[4]);
645     $stmt->bindParam("photo_6", $scheda-> photo[5]);
646     $stmt->bindParam("photo_7", $scheda-> photo[6]);
647     $stmt->bindParam("photo_8", $scheda-> photo[7]);
648     $stmt->bindParam("photo_9", $scheda-> photo[8]);
649     $stmt->bindParam("photo_10", $scheda-> photo[9]);
650 }
651
652     $stmt->execute();
653
654 // $result = $stmt->fetchObject();
655 // echo json_encode($result);
656
657     echo json_encode($db->lastInsertId());
658     $db = null;
659
660
661 } catch(PDOException $e) {
662 /**
663 *     error_log($e->getMessage(), 3, '/var/tmp/php.log');
664 */
665     echo '{"error":{"text":' . $e->getMessage() . '} }';
666 }
667 }
668
669 function mailto() {
670 /**
671 * funzione per la spedizione di email
```

## index.php

```
672 */
673 $request = Slim::getInstance()->request();
674 $mail = json_decode($request->getBody());
675 $to = $mail->to;
676 $subject = $mail->subject;
677 $body = $mail->body;
678 try {
679     mail($to, $subject, $body);
680
681     echo '{"ok": {"text": "Email successfully sent"} }';
682 } catch(PDOException $e) {
683     echo '{"error": {"text": "' . $e->getMessage() . '}}';
684 }
685
686 }
687
688
689
690
691 function getConnection() {
692     $dbhost="sql.dntppe.net";
693     $dbuser="dntpene47150";
694     $dbpass="dntp59296";
695     $dbname="dntpene47150";
696     $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser, $dbpass);
697     $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
698     return $dbh;
699 }
700
701 ?>
```

## **BIBLIOGRAFIA**

Di Domenico G., Ronzoni S., *Guida alla prova scritta di Estimo per gli istituti tecnici per geometri e periti agrari*, Arnaldo Mondadori, 1995.

Franchi D., Ragagnin G.C., *Esercizi stime modelli prontuario*, Editore Bulgarini Firenze, 2002.

Medici G., *Elementi di estimo civile, rurale e catastale*, Edagricole, 1977.

Simonotti M., *Metodi di valutazione immobiliare*, Dario Flaccovio Editore, Palermo, 2006.

Simonotti M., *Stime catastali e standard internazionali*, in "Estimo e territorio" , n. 3, 2008.

Simonotti M., *Stime catastali e standard internazionali*, in "Estimo e territorio" , n. 3, 2008.

## **SITI INTERNET**

<http://www.w3schools.com/>

<http://www.html.it/guide/guida-apache-cordova/>

<http://docs.phonegap.com/>

<http://api.jquery.com/>

<http://jquerymobile.com/>

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

<http://www.webtoolkit.info/>

<http://stackoverflow.com/>

<https://developers.google.com/maps/?hl=it>

<http://php.net/docs.php>

<http://www.slimframework.com/>

<http://dev.mysql.com/doc/>

<http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/Istruzioni+determinazione/consistenza.pdf>

[http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/tabella+guide/guida\\_schede.pdf](http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/tabella+guide/guida_schede.pdf)

[http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/Guida+alla+consultazione+delle+quotazioni+OMI/guida\\_cons\\_quotOMI\\_20130724.pdf](http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/Guida+alla+consultazione+delle+quotazioni+OMI/guida_cons_quotOMI_20130724.pdf)

[http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/Il+manuale+della+banca+dati+OMI/Manuale\\_OMI\\_luglio2009\\_rev\\_logo.pdf](http://www.agenziaentrate.gov.it/wps/file/Nsilib/Nsi/Documentazione/omi/Manuale+e+guide/Il+manuale+della+banca+dati+OMI/Manuale_OMI_luglio2009_rev_logo.pdf)