



Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea Triennale in Ingegneria Informatica

tesi di laurea

# Analisi e sviluppo di un Webgis per il monitoraggio urbano

**Relatore:** Massimo Rumor

**Laureando:** Riccardo Braga

ver.α – 27 marzo 2011

---

# Indice

<b>1</b>	<b>Descrizione del progetto</b>	<b>1</b>
1.1	Introduzione . . . . .	1
1.2	Azienda ospitante . . . . .	2
1.3	Analisi di mercato . . . . .	2
<b>2</b>	<b>Progettazione</b>	<b>5</b>
2.1	Casi d'uso . . . . .	5
2.1.1	Utenti . . . . .	5
2.1.2	Amministratori . . . . .	7
2.2	Architettura applicativa . . . . .	8
2.2.1	AJAX . . . . .	8
2.3	Basi di dati . . . . .	10
2.3.1	Schema concettuale e logico . . . . .	10
2.3.2	Schema fisico . . . . .	13
<b>3</b>	<b>Realizzazione software</b>	<b>19</b>
3.1	Strumenti utilizzati . . . . .	19
3.1.1	PHP . . . . .	19
3.1.2	Zend Framework . . . . .	20
3.1.3	PostgreSQL . . . . .	21
3.1.4	Google Maps . . . . .	21
3.1.5	JQuery . . . . .	21
3.2	Risultato ottenuto . . . . .	22
3.2.1	Pagina iniziale . . . . .	22
3.2.2	Login di un utente . . . . .	23
3.2.3	Inserimento di una nuova segnalazione . . . . .	23
3.2.4	Ricerca di una segnalazione . . . . .	24
3.2.5	Informazioni sul servizio . . . . .	25
<b>4</b>	<b>Sviluppi futuri e conclusione</b>	<b>27</b>
	<b>Bibliografia</b>	<b>29</b>



# Capitolo 1

## Descrizione del progetto

### 1.1 Introduzione

Obiettivo del lavoro durante il periodo di tirocinio è la realizzazione di un'applicazione Web che permetta a un utente di poter segnalare e registrare nella relativa base di dati, problemi e situazioni di disagio nell'ambiente cittadino.

Con la crescente diffusione dell'ICT, sempre più persone possiedono al giorno d'oggi una connessione internet e, con il trascorrere del tempo, tutte le operazioni per le quali una volta si doveva rivolgersi all'ufficio specifico, oggi servizi quali operazioni bancarie, pagamenti postali ecc., sono effettuabili comodamente da casa propria. Lo scopo di questo progetto si basa per l'appunto su questo concetto: permette al cittadino, nel caso noti un problema nell'ambiente urbano un problema tale per cui è necessario un intervento alle autorità, di segnalare qualsiasi tipo di disagio tramite un comune pc e una connessione ad internet, senza quindi doversi recare in un ufficio e compilare moduli cartacei, ma semplicemente compilando un form in un sito internet.

Questa applicazione è accessibile da ogni utente online e richiede esclusivamente un browser (internet explorer, firefox ecc.) e una comune connessione a internet. Vengono inoltre utilizzate le funzionalità di GoogleMaps per facilitare e rendere più precise le segnalazioni registrate sia per gli utenti per indicare con maggiore precisione il punto esatto del problema, sia per gli amministratori, che possono verificare e segnalare alle autorità competenti il punto preciso dove operare senza dilungarsi troppo. La realizzazione di questa applicazione è motivata dalla tempistività e rapidità che vengono offerte dal web, ormai in continua espansione e sempre più utilizzato, rispetto ai vecchi canali di comunicazione per effettuare richieste e segnalazioni di disagio alla pubblica amministrazione di carattere urbano. Le segnalazioni possono essere effettuate da utenti registrati e non, con la differenza che questi ultimi non possono commentare altre segnalazioni e devono indicare nome e cognome ogni volta che effettuano una nuova segnalazione. In ogni segnalazione sarà presente il punto esatto della segnalazione, indicato tramite la mappa visualizzabile dal form, il quale indicherà

inoltre via e città della segnalazione, che categoria e tipo di problema si vuole segnalare, l'oggetto della segnalazione, un'eventuale descrizione aggiuntiva con la quale aggiungere particolari che non sono evidenziabili dalla scelta del tipo di problema fra i presenti e un'eventuale immagine in formato digitale, caricata direttamente dall'utente per evidenziare in maniera ancora migliore il problema. Oltre agli utenti, saranno presenti anche degli amministratori, i cui compiti saranno quelli di validare e aggiornare lo stato delle segnalazioni registrate, inoltrarle all'ufficio di competenza, moderare i commenti degli utenti riguardanti le varie segnalazioni e assistere gli utenti in caso questi abbiano problemi con le funzionalità del servizio. Per memorizzare gli utenti registrati, le segnalazioni, gli amministratori, i commenti riguardanti le segnalazioni e gli aggiornamenti di stato viene progettato e realizzato un database, sfruttando come RDBMS PostgreSQL con le funzionalità georeferenziali aggiuntive offerte da PostGIS. La tesi è stata organizzata in quattro capitoli nei quali nel primo vengono descritti gli strumenti utilizzati per realizzare il progetto, nel secondo la progettazione del database, nel terzo una descrizione delle funzionalità del sito in ogni pagina e nel quarto vengono descritti possibili sviluppi futuri per migliorare e aggiornare il software.

## 1.2 Azienda ospitante

Lo svolgimento del tirocinio è avvenuto presso la sede operativa dell'azienda 3DGIS srl, una giovane società costruita su un'esperienza pluriennale nel campo della progettazione e dello sviluppo di sistemi informativi territoriali e in generale di soluzione nell'area ITC e GeoITC. E' la combinazione di diverse professionalità: ingegneri informatici, grafici e architetti esperti in problematiche di dominio, con esperienza di analisi, progettazione e sviluppo di applicazioni Gis e Web Gis 2D e 3D. I clienti dell'azienda ospitante sono per lo più enti pubblici di medie/piccole dimensioni che intuiscono la necessità di dotarsi di strumenti di creazione e/o gestione delle proprie banche dati e informazioni territoriali, ma che spesso si scontrano con soluzioni preconfezionate, non adatte alle loro realtà e piuttosto onerose soprattutto a regime.

## 1.3 Analisi di mercato

Al fine di comprendere al meglio le problematiche e le funzioni dell'applicativo stesso sono state svolte, prima di iniziare la fase di progettazione, delle ricerche sul web per cercare e analizzare prodotti simili già esistenti e disponibili. La ricerca ha portato a un'analisi maggiormente approfondita dei seguenti servizi:

- iris venezia (<http://iris.comune.venezia.it/>);
- urbis; (<http://www.sistemait.com/ita/prodotti/monitoraggio-e-infomobilita/urbis-urban-information-system/>)

	Inserimento	Profilazione utenti	Visualizzazione	Commenti	Ricerca
iris venezia	X		X	X	X
urbis	X	X (obbligatoria)	X	X	X
urbsit webgis palatino			X		X
partecipaMI	X	X (obbligatoria)	X	X	X
mappaTO			X		X
ePart	X	X (facoltativa)	X	X	X

Tabella 1.1: Analisi di mercato: funzionalità degli applicativi attualmente esistenti

- urbsit webgis palatino; (<http://webgis.igag.cnr.it/palatino/>)
- partecipaMI; (<http://www.manifestopermilano.partecipami.it/>)
- mappaTO; (<http://www.comune.torino.it/servizionline/mappaTo/>)
- ePart. (<http://www.epart.it/>)

In figura 1.1 vengono evidenziate le funzionalità di ogni applicativo e dall'analisi dei siti, si può notare come la struttura generale sia simile: la sezione di inserimento di una segnalazione si basa sempre sul posizionamento di un marcatore su una mappa, il quale salverà i dati geospaziali del problema da segnalare, sull'inserimento dei dati geografici del problema quali comune, via, indirizzo e numero civico, e sull'inserimento dei dati anagrafici dell'utente segnalatore, il quale può decidere di rimanere anonimo in fase di visualizzazione.

La profilazione degli utenti non è sempre gestita, e quando è presente ha lo scopo principale di differenziare gli utenti concedendo maggiori privilegi e minori campi da compilare agli utenti che si registrano nella base di dati dell'applicativo. A prescindere però da questo, tutti gli utenti possono tranquillamente inserire segnalazioni senza nessun obbligo di registrazione o simili.

La visualizzazione delle segnalazioni è praticamente identica in ogni applicativo: viene visualizzata la mappa principale con tutte le segnalazioni presenti visualizzate come un marcatore e posizionate in base alla geometria registrata. Cliccando sul suddetto marcatore si possono visualizzare i dati relativi alla segnalazione stessa quali utente segnalatore, oggetto, data ecc.

Tramite l'analisi dei sopracitati applicativi e delle loro funzionalità è stato possibile scegliere:

- le possibili tipologie di problemi che riguarderanno le segnalazioni;

- i campi necessari per la registrazione di ogni segnalazione in maniera tale da avere quante più informazioni utili possibili;
- le metodologie di visualizzazione dei marcatori nella mappa;
- le statistiche da calcolare in base alle segnalazioni presenti e alle loro informazioni;
- la disposizione degli oggetti visivi dell'applicativo e loro strutturazione e differenziazione in base alle funzioni che dovranno essere svolte;
- privilegi e limiti di utenti registrati, utenti non registrati e amministratori.



# Capitolo 2

# Progettazione

## 2.1 Casi d'uso

### 2.1.1 Utenti

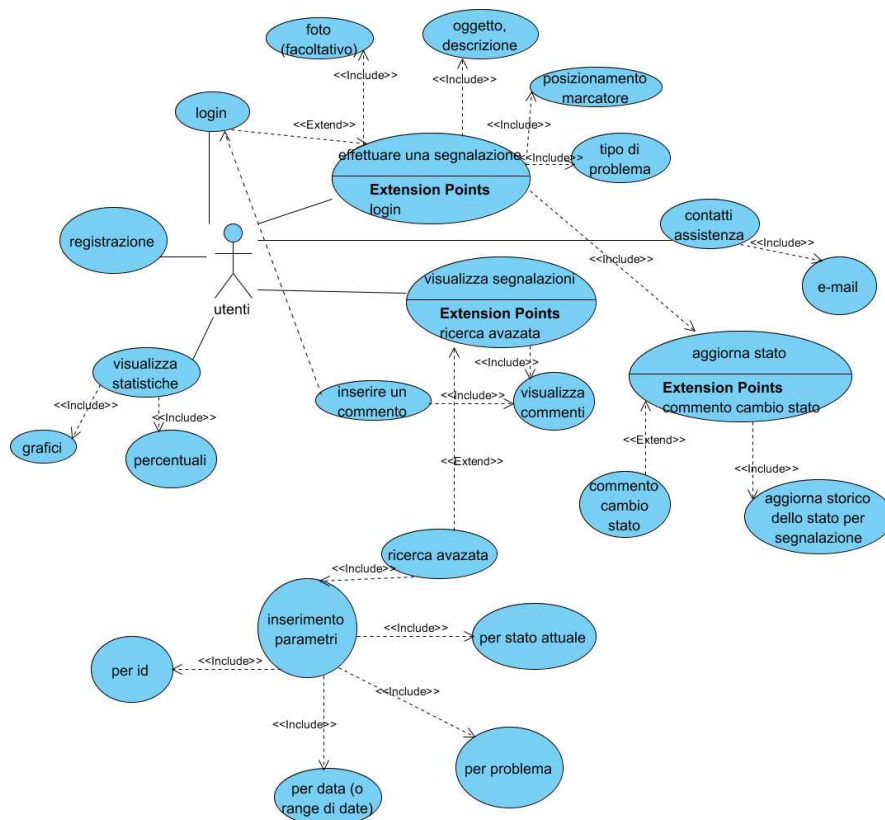


Figura 2.1: Casi d'uso per gli utenti

In questo paragrafo verranno descritti i vari casi d'uso che si ottengono dalle interazioni dell'utente con l'applicazione.

Un utente generico ha la possibilità di registrarsi in modo tale da poter conservare i propri dati anagrafici nell'eventualità volesse effettuare altre segnalazioni in futuro. Ciononostante la registrazione non è obbligatoria, ciò per non vincolare degli utenti occasionali a una registrazione che magari non desiderano effettuare agli utenti, infatti la successiva operazione di login può essere omessa per poter procedere direttamente al form per l'inserimento della segnalazione, dove verranno immesse informazioni quali la posizione esatta del caso (tramite Google Maps), il tipo di problema, un oggetto, una descrizione (facoltativa), e una o più foto (facoltativa). C'è da tenere inoltre presente che nel caso di un utente registrato, non sarà possibile inserire il nome e il cognome del segnalatore, in quanto verranno automaticamente utilizzati i relativi dati anagrafici presenti nella base di dati, mentre un utente non registrato dovrà obbligatoriamente inserire anche questi dati.

E' possibile inoltre per gli utenti visualizzare tutte le segnalazioni effettuate fino a quel momento, tramite la mappa principale dove verranno sistemati tutti i relativi marcatori, oppure, tramite una ricerca avanzata, è possibile visualizzare solo determinate segnalazioni che rispettano delle condizioni imposte dall'utente. Queste condizioni riguarderanno i dati principali della segnalazioni quali id, problema, stato attuale e data e saranno inserite dall'utente stesso in base alla ricerca che vuole effettuare.

E' inoltre possibile lasciare dei commenti riguardanti una specifica segnalazione, i quali dovranno essere prima validati prima di poter essere visualizzati dal resto dell'utenza; tali commenti però potranno essere introdotti solo da utenti registrati che hanno effettuato con successo l'operazione di login, in caso contrario potranno solo visualizzare quelli validati. Viene inoltre data l'opportunità all'utenza di mandare mail o comunicazioni varie agli amministratori di sistema per aiuti, dubbi ecc. sulle funzionalità del sito.

Ricapitolando i privilegi concessi agli utenti non registrati sono:

- inserimento di una segnalazione con i propri dati anagrafici;
- visualizzazione delle segnalazioni presenti nella base di dati;
- visualizzazione dei commenti per ogni segnalazione;
- visualizzazione dello stato di ogni segnalazione;
- richiedere aiuti o consigli dagli amministratori;
- visualizzazione delle statistiche.

Mentre un utente registrato può, oltre ai privilegi indicati sopra:

- modificare i dati del proprio profilo;

- inserire un commento per una segnalazione;
- inserire una nuova segnalazione senza dover reinserire i dati anagrafici.

Oltre a tutto ciò gli utenti potranno visualizzare delle statistiche riguardanti l'applicativo e le segnalazioni registrate (funzionalità futura).

### 2.1.2 Amministratori

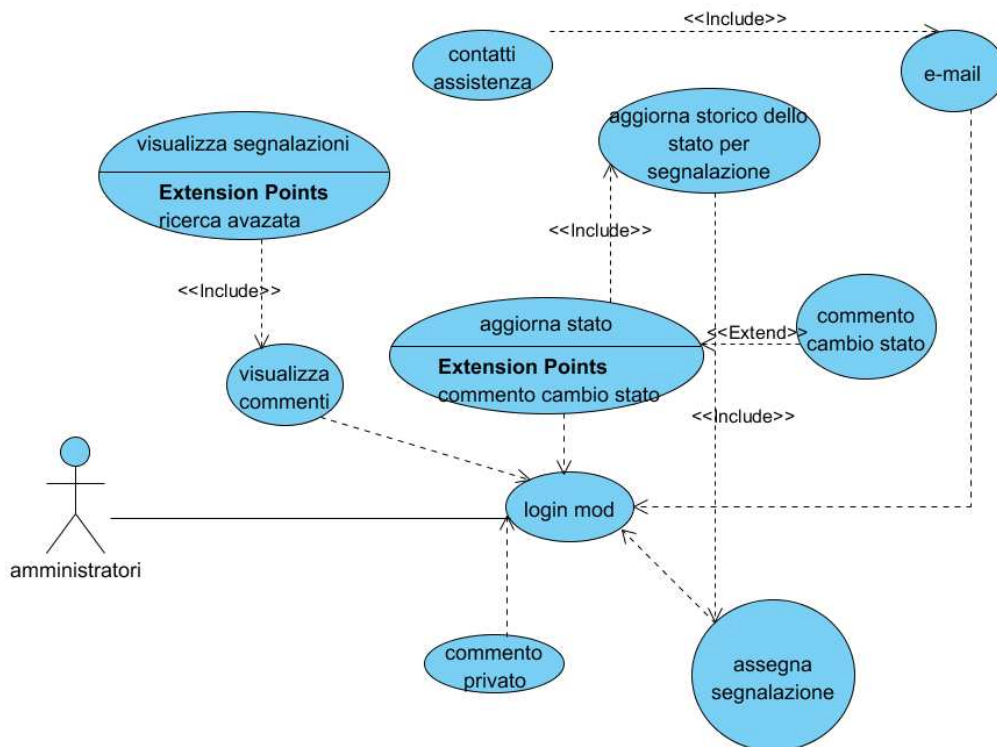


Figura 2.2: Casi d'uso per gli amministratori

In questo paragrafo verranno descritti i vari casi d'uso che si ottengono dalle interazioni degli amministratori con l'applicazione.

Gli amministratori del sistema per effettuare ogni tipo di operazione, devono obbligatoriamente essere registrati nella base di dati e aver effettuato il login con successo.

L'operazione principale degli amministratori è quella di aggiornare gli stati delle segnalazioni presenti nel sistema, di inoltrarle all'ufficio di competenza e eventualmente passarle a un altro amministratore; possono inoltre aggiungere una descrizione per giustificare o aggiungere particolari al cambio di stato effettuato.

Altro importante compito è quello di moderare i commenti rilasciati dagli utenti registrati, validandoli e rendendoli visibili a tutti oppure lasciandoli oscurati in caso non vengono considerati accettabili. Vengono inoltre definiti solo per gli amministratori dei commenti privati, ovvero dei commenti che solo gli amministratori possono vedere e visualizzabili attraverso lo stesso form; un amministratore dunque, oltre ai commenti degli utenti, vedrà anche i commenti degli amministratori mentre gli utenti in nessun caso vedranno i commenti degli amministratori.

Gli amministratori devono inoltre rispondere alle richieste e ai dubbi degli utenti che sono registrati nella base di dati (funzionalità futura).

Ricapitolando, gli amministratori hanno i seguenti compiti:

- registrare l'aggiornamento dello stato delle segnalazioni;
- validare o rifiutare la visualizzazione dei commenti degli utenti;
- rispondere alle richieste degli utenti;
- rilasciare commenti privati nelle segnalazioni per eventuali comunicazioni con gli altri amministratori.

## 2.2 Architettura applicativa

L'architettura su cui si baserà il progetto consiste nel modello 3-tier, che rappresenta una particolare architettura in cui il progetto viene suddiviso in tre diversi moduli dedicati rispettivamente alla interfaccia utente (user logic), alla logica funzionale (business logic) e alla gestione dei dati persistenti (data logic). L'architettura 3-tier è un'evoluzione dell'architettura client/server in cui, base di dati e controllo della logica erano gestiti esclusivamente dal server e, di conseguenza, il client era obbligatoriamente legato al server. Con l'inserimento di un terzo componente detto business tier, il client viene svincolato dal server e tutte le richieste al server passano prima al business tier tramite chiamate http e quest'ultimo restituirà al client codice xml, html, json o altri tipi.

### 2.2.1 AJAX

L'acronimo AJAX sta per Asynchronous Javascript and XML e non si tratta di una nuova tecnologia o di un'invenzione bensì di un concetto utilizzato per sviluppare applicativi avanzati e particolari quali, nel caso in uso, Google Maps. Sviluppare applicativi web con AJAX si basa su uno scambio di dati in background tra il client e il motore AJAX, che viene caricato alla prima apertura del sito, in modo da svincolare il client dal server. Ciò per permettere un aggiornamento dinamico della pagina, senza dover quindi dover effettuare un nuovo ricaricamento della pagina.

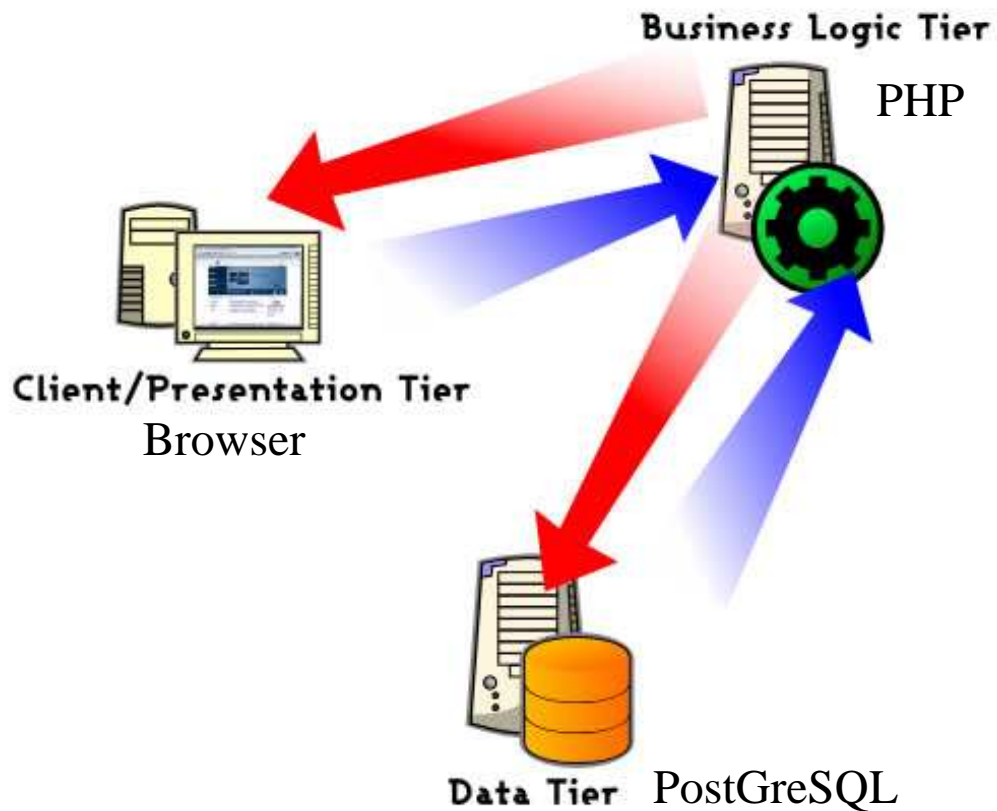


Figura 2.3: Architettura 3-tier

Nel dettaglio, ogni richiesta del client, che normalmente genererebbe una richiesta HTTP, diviene una chiamata in Javascript per il motore AJAX. In caso di richieste che non richiedono l'utilizzo del server, per esempio la validazione dei dati in un form, queste vengono gestite direttamente da AJAX. In caso contrario, quindi se è necessario ricevere o mandare dati al server, AJAX rende queste chiamate asincrone, in maniera da evitare all'utente di attendere la risposta senza poter effettuare altre operazioni, come normalmente avverrebbe con chiamate sincrone.

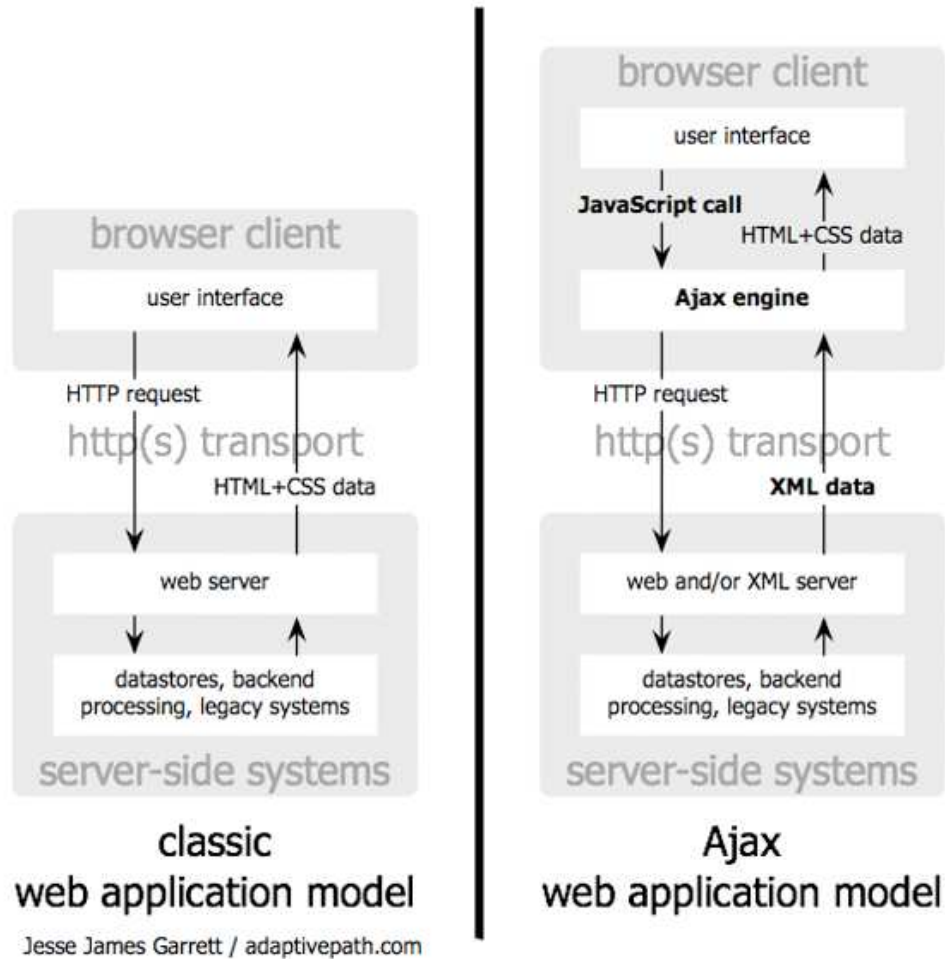


Figura 2.4: Schema funzionamento ajax

## 2.3 Basi di dati

### 2.3.1 Schema concettuale e logico

La basi di dati si basa fundamentalmente sul salvataggio e la gestione delle segnalazioni, le quali sono la base portante dell applicativo e tramite i dati geometrici salvati per ognuna verrà visualizzata sulla mappa la rispettiva posizione.

Ogni segnalazione può essere effettuata da uno e un solo utente e viene identificata da:

- un id progressivo (chiave);
- l'utente che ha effettuato la segnalazione (registrato o meno);



- nome e cognome;
- residenza;
- indirizzo e-mail;
- username e password per l'accesso;
- grado.

Esiste inoltre la possibilità per ogni utente registrato di commentare le segnalazioni presenti e pubbliche con un apposito form, e ogni commento dovrà essere validato da un amministratore perchè possa essere visualizzato da tutti. Ogni segnalazione può avere più commenti mentre un commento può riguardare una e una sola segnalazione e viene definito con:

- un id progressivo (chiave);
- la segnalazione da commentare;
- il commento;
- l'utente che lascia il commento;
- validazione del commento;
- la data e l'ora di registrazione.

Solo gli amministratori hanno la possibilità di modificare il campo di validazione.

Questi ultimi vengono anche loro registrati nella base di dati e hanno il compito di gestire e aggiornare l'applicativo. Per ogni amministratore si tiene conto di:

- un id progressivo (chiave);
- nome e cognome;
- residenza;
- indirizzo e-mail;
- username e password per l'accesso;
- grado.

Gli utenti comuni hanno sempre grado -1.

Gli amministratori hanno il compito di aggiornare lo stato di ogni segnalazione registrata e per ogni segnalazione si vuole conservare uno storico delle varie modifiche dello stato di conseguenza una segnalazione potrà avere più stati mentre uno stato riguarderà una e una sola segnalazione. Per ogni stato si vuole tenere traccia di:



- un id progressivo (chiave);
- la segnalazione in esame;
- l'amministratore che effettua il cambio stato;
- il nuovo stato;
- ufficio a cui viene inoltrato;
- commento (facoltativo);
- la data e l'ora di aggiornamento.

Esiste inoltre la possibilità per gli utenti registrati di chiedere domande e consigli agli admin del sito tramite un apposito. Ogni comunicazione viene effettuata da uno e un solo utente e da uno e un solo amministratore che risponde e si vuole tenere conto di:

- un id progressivo (chiave);
- un utente;
- un amministratore;
- il testo della richiesta;
- oggetto della comunicazione;
- tipo della comunicazione.

Vengono infine messe a disposizione delle entità quali comune, categoria, problema e grado, le quali impediscono al sistema e agli utenti di registrare valori non pertinenti o assenti.

Da questa base concettuale si evidenziano l'assenza di ridondanze, di generalizzazioni e attributi multi-valore e non si ritengono necessari eventuali accorpamenti e/o partizionamenti dei concetti sopra rappresentati; di conseguenza lo schema logico sarà una semplice traduzione dello schema concettuale sopra riportato, come mostrato in figura.

### 2.3.2 Schema fisico

Dallo schema sopra riportato, si può procedere con la compilazione del codice SQL sotto riportato, affinché possa essere gestito tramite PostgreSQL.

```
CREATE TABLE utente (  
  id          SERIAL NOT NULL,  
  nome       varchar(255) NOT NULL,
```

```
cognome    varchar(255) NOT NULL,  
residenza  varchar(255) NOT NULL,  
email      varchar(255) NOT NULL,  
username   varchar(255) NOT NULL,  
password   varchar(255) NOT NULL,  
grado      int4 NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE segnalazione (  
  id          SERIAL NOT NULL,  
  utente      int4 NOT NULL,  
  comune      varchar(255) NOT NULL,  
  problema    int4 NOT NULL,  
  categoria   int4 NOT NULL,  
  oggetto     varchar(255) NOT NULL,  
  descrizione varchar(255),  
  data        timestamp NOT NULL,  
  foto        varchar(255),  
  geom        varchar(255) NOT NULL,  
  indirizzo   varchar(255) NOT NULL,  
  cap         varchar(255) NOT NULL,  
  ncivico     varchar(255) NOT NULL,  
  privato     bool NOT NULL,  
  PRIMARY KEY (id));
```

```
CREATE TABLE commento (  
  id          SERIAL NOT NULL,  
  utente      int4 NOT NULL,  
  segnalazione int4 NOT NULL,  
  data        timestamp NOT NULL,  
  commento    varchar(255) NOT NULL,  
  approvato   bool NOT NULL,  
  PRIMARY KEY (id));
```

```
CREATE TABLE comune (  
  codice_istat varchar(255) NOT NULL,  
  nome          varchar(255) NOT NULL,  
  PRIMARY KEY (codice_istat));
```

```
CREATE TABLE problema (  
  id          int4 NOT NULL,  
  categoria   int4 NOT NULL,  
  nome        varchar(255) NOT NULL,
```

```
PRIMARY KEY (id,  
categoria));
```

```
CREATE TABLE categoria (  
id SERIAL NOT NULL,  
nome varchar(255) NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE commento_privato (  
id SERIAL NOT NULL,  
data timestamp NOT NULL,  
commento varchar(255) NOT NULL,  
segnalazione int4 NOT NULL,  
amministratore int4 NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE comunicazioni (  
id SERIAL NOT NULL,  
data timestamp NOT NULL,  
testo varchar(255) NOT NULL,  
amministratore int4 NOT NULL,  
utente int4 NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE stato (  
id SERIAL NOT NULL,  
segnalazione int4 NOT NULL,  
amministratore int4 NOT NULL,  
stato int4 NOT NULL,  
data_agg timestamp NOT NULL,  
commenti varchar(255),  
ufficio int4 NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE tipo_stato (  
id SERIAL NOT NULL,  
nome varchar(255) NOT NULL,  
simbolo varchar(255) NOT NULL,  
PRIMARY KEY (id));
```

```
CREATE TABLE ufficio (  
id SERIAL NOT NULL,  
nome varchar(255) NOT NULL,
```

```
PRIMARY KEY (id));

CREATE TABLE amministratore (
  id          SERIAL NOT NULL,
  nome        varchar(255) NOT NULL,
  cognome     varchar(255) NOT NULL,
  residenza   varchar(255) NOT NULL,
  email       varchar(255) NOT NULL,
  username    varchar(255) NOT NULL,
  password    varchar(255) NOT NULL,
  grado       int4 NOT NULL,
  PRIMARY KEY (id));

CREATE TABLE grado (
  id          SERIAL NOT NULL,
  nome        varchar(255) NOT NULL,
  PRIMARY KEY (id));

ALTER TABLE segnalazione ADD CONSTRAINT FKsegnalazio347499
  FOREIGN KEY (utente) REFERENCES utente (id);
ALTER TABLE commento ADD CONSTRAINT FKcommento337251
  FOREIGN KEY (utente) REFERENCES utente (id);
ALTER TABLE commento ADD CONSTRAINT FKcommento253261
  FOREIGN KEY (segnalazione) REFERENCES segnalazione (id);
ALTER TABLE segnalazione ADD CONSTRAINT FKsegnalazio830173
  FOREIGN KEY (comune) REFERENCES comune (codice_istat);
ALTER TABLE problema ADD CONSTRAINT FKproblema818224
  FOREIGN KEY (categoria) REFERENCES categoria (id);
ALTER TABLE segnalazione ADD CONSTRAINT FKsegnalazio463549
  FOREIGN KEY (problema, categoria) REFERENCES problema (id, categoria);
ALTER TABLE commento_privato ADD CONSTRAINT FKcommento_p256150
  FOREIGN KEY (segnalazione) REFERENCES segnalazione (id);
ALTER TABLE stato ADD CONSTRAINT FKstato920677
  FOREIGN KEY (segnalazione) REFERENCES segnalazione (id);
ALTER TABLE stato ADD CONSTRAINT FKstato921239
  FOREIGN KEY (stato) REFERENCES tipo_stato (id);
ALTER TABLE stato ADD CONSTRAINT FKstato545215
  FOREIGN KEY (ufficio) REFERENCES ufficio (id);
ALTER TABLE amministratore ADD CONSTRAINT FKamministra685893
  FOREIGN KEY (grado) REFERENCES grado (id);
ALTER TABLE commento_privato ADD CONSTRAINT FKcommento_p743419
  FOREIGN KEY (amministratore) REFERENCES amministratore (id);
ALTER TABLE comunicazioni ADD CONSTRAINT FKcomunicazi325226
```

---

```
    FOREIGN KEY (amministratore) REFERENCES amministratore (id);
ALTER TABLE stato ADD CONSTRAINT FKstato892698
    FOREIGN KEY (amministratore) REFERENCES amministratore (id);
ALTER TABLE comunicazioni ADD CONSTRAINT FKcomunicazi27873
    FOREIGN KEY (utente) REFERENCES utente (id);
```



## Capitolo 3

# Realizzazione software

### 3.1 Strumenti utilizzati

La scelta degli strumenti di programmazione utilizzati per la realizzazione del progetto è stata fatta basandosi su software FOSS, acronimo per Free Open Source Software, ovvero software liberi e open, prevalentemente usato anche nell'azienda ospitante, la quale ha per l'appunto fornito consigli sui software più adatti per ogni parte del progetto in esame. I vantaggi fondamentali di tale tipologia di software si può riassumere in:

- disponibilità del codice sorgente;
- aggiornamenti continui e gratuiti;
- ampia disponibilità di risorse informative e di supporto online tramite le community.

#### 3.1.1 PHP

Il PHP, acronimo ricorsivo di PHP: Hypertext Preprocessor e ideato originariamente da Rasmus Lerdorf nel 1994, è un linguaggio di programmazione progettato principalmente per la costruzione di siti Web dinamici e può essere incorporato all'interno del codice HTML. Appartiene ad una classe di linguaggi detti middleware. Questi linguaggi lavorano a stretto contatto con il server Web per interpretare le richieste fatte dal client, elaborarle, interagendo anche con altri programmi sul server, e producono l'output che il server Web fornirà al browser.

PHP è un linguaggio la cui funzione fondamentale è quella di produrre codice HTML, che è quello dal quale sono formate le pagine Web. Ma, poichè PHP è un linguaggio di programmazione, abbiamo la possibilità di analizzare diverse situazioni (l'input degli utenti, i dati contenuti in un database) e di decidere, di conseguenza, di produrre codice HTML condizionato ai risultati dell'elaborazione. La sintassi del PHP è simile a quella di C, Java e Perl e ciò permette ai programmatori che già conoscono questi linguaggi un più semplice approccio.

Uno dei punti di forza del PHP è inoltre la capacità di connettersi e interagire facilmente con i più diffusi database (PostgreSQL, Oracle, MySQL, IBM DB2, Sybase, Informix, etc).

Oltre ai motivi sopracitati per cui viene scelto PHP come linguaggio server-side rispetto ad altri attualmente esistenti, va evidenziata la sua semplicità e il fatto che viene considerato il linguaggio principale in fase di sviluppo web.

### 3.1.2 Zend Framework

Zend Framework è un web application framework open source creato per semplificare e rendere più efficace sotto ogni punto di vista la produzione di applicativi e servizi web PHP-based. Zend Framework contiene un insieme di componenti riutilizzabili che rispondono nella sostanza a tutti i requisiti richiesti per un completo applicativo web-based: questi componenti sono scritti interamente in PHP seguendo le migliori pratiche della programmazione ad oggetti.

Zend Framework offre molti vantaggi rispetto ad altri framework presenti nel mercato attuale, vantaggi che lo rendono la scelta principale per la realizzazione del progetto in esame. Tali vantaggi sono:

- Rapidità e semplicità di sviluppo;
- Design moderno;
- Facilità di apprendimento;
- Documentazione completa;
- Community driven.

Zend Framework è caratterizzato da un robusto componente Model-View-Controller (MVC) per strutturare l'applicativo secondo le specifiche dettate dall'omonimo design pattern, uno per la gestione della ricerca, uno per l'autenticazione, uno per la gestione della localizzazione, un altro per la generazione dinamica di PDF, e molto altro ancora. Proprio a causa di questa sua caratteristica modulare è usuale riferirsi a questo framework come ad una libreria di componenti.

Ciò che differenzia Zend Framework da altri framework PHP è la sua particolare struttura a compartimenti stagni: l'indipendenza delle sue componenti permette di utilizzare solo il necessario per il proprio progetto, ma allo stesso tempo, se utilizzate insieme, esse si integrano in un substrato incredibilmente potente e semplice da imparare, indirizzando il lavoro dello sviluppatore verso soluzioni basate sui principi di riutilizzo del codice, di estendibilità, leggibilità e manutenzione.

Zend Framework è rilasciato sotto licenza BSD: questo permette al framework di essere inserito nella maggior parte dei progetti, siano questi open source o proprietari, con le minori restrizioni possibili per gli utilizzatori.



### 3.1.3 PostgreSQL

PostgreSQL è un completo e potente database relazionale ad oggetti rilasciato con licenza libera, diretto concorrente di altri famosi DBMS quali MySQL, Oracle, Microsoft Access ecc.

Viene scelto questo particolare RDBMS per il progetto in esame non tanto per le sue caratteristiche e funzionalità native, ma per il fatto che può essere integrato con un pacchetto conosciuto come PostGis che aggiunge supporto e funzionalità per oggetti geospaziali, fornendo i tipi di dati specificati negli standard dell'Open Geospatial Consortium e permettendo a PostgreSQL di poter gestire tali oggetti. PostgreSQL dunque può essere sfruttato come database spaziale per sistemi informativi territoriali (GIS). La natura del progetto si basa appunto sulla gestione di dati geospaziali, i quali conterranno la posizione delle segnalazioni registrate dagli utenti; pertanto tramite l'estensione PostGis, PostgreSQL risulta essere la migliore scelta per la gestione di dati geospaziali.

### 3.1.4 Google Maps

Google Maps è un servizio di Google che fornisce una semplice e potente tecnologia di mappatura, insieme a informazioni sulle attività commerciali locali, come indirizzi, dati di contatto e indicazioni stradali. Google ha lanciato le API di Google Maps nel 2005 per permettere agli sviluppatori di integrare Google Maps all'interno dei propri siti web, in maniera completamente gratuita e attualmente senza alcuna pubblicità aggiuntiva. Mentre all'inizio le API erano esclusivamente Javascript, ora sono state estese anche per applicativi Flash, per servizi di recupero di mappe statiche e per servizi geospaziali, quali calcolo di itinerari stradali. Oltre 350.000 siti web sfruttano le API di Google Maps, rendendole le più usate nell'ambito degli applicativi web-based. In definitiva è proprio per tale diffusione, per la semplicità delle API e per le funzionalità che vengono offerte, perfettamente adatte agli scopi del progetto, i motivi fondamentali per cui viene scelto il servizio offerto da Google per la gestione delle mappe e dei dati spaziali.

### 3.1.5 JQuery

jQuery è un framework codificato in Javascript il cui scopo è quello di rendere il codice più sintetico e di limitare al minimo l'estensione degli oggetti globali per ottenere la massima compatibilità con altre librerie. Da questo principio è nata una libreria in grado di offrire un'ampia gamma di funzionalità, che vanno dalla manipolazione degli stili CSS e degli elementi HTML, agli effetti grafici per passare a comodi metodi per chiamate AJAX cross-browser. Il tutto, appunto, senza toccare nessuno degli oggetti nativi JavaScript. JQuery quindi risulta essere un elemento importante per la realizzazione del progetto in quanto consente una facile realizzazione di quelle che sarebbero complesse istruzioni javascript per eseguire chiamate o richieste al server sfruttando AJAX, e perme-

tte la creazione di dialog e form dinamici che permettono maggiore velocità e praticità all'applicativo in quanto non avviene alcun ricaricamento della pagina visualizzata.

## 3.2 Risultato ottenuto

In questo capitolo vengono ora descritte le funzionalità base del progetto dal punto di vista dell'utente, la quale non avrà alcun accesso alle sezioni amministrative o tantomeno ai privilegi a loro concessi.

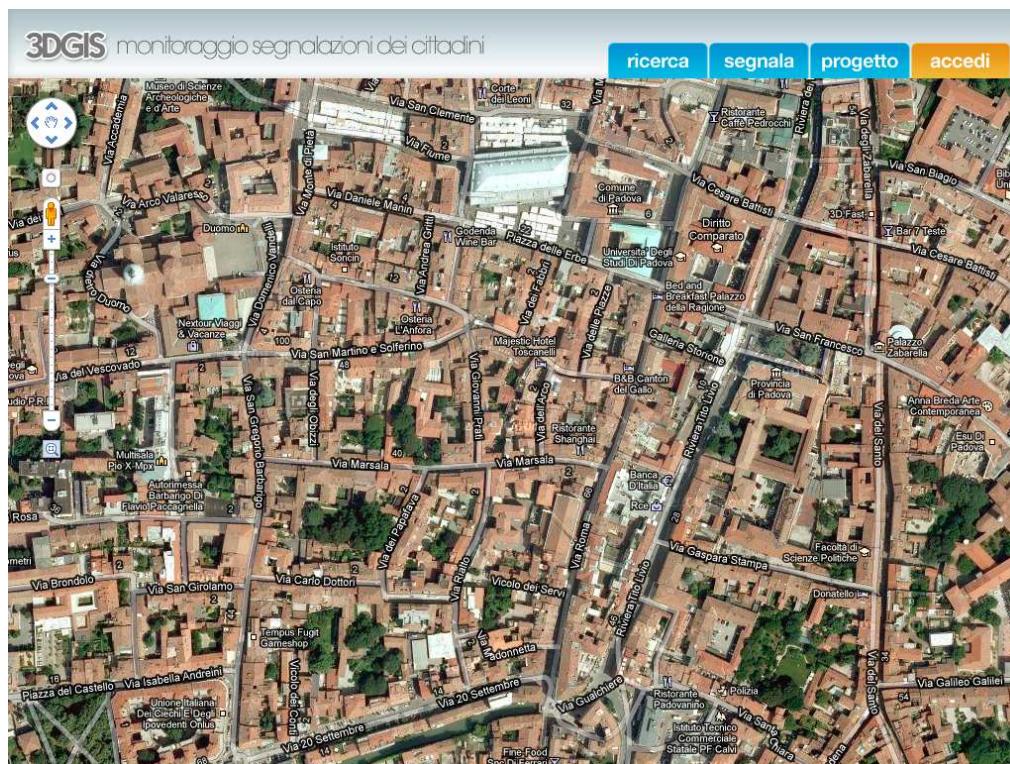


Figura 3.1: Pagina iniziale

### 3.2.1 Pagina iniziale

La pagina d'apertura presenta in alto a destra i pulsanti per segnalare un problema, effettuare ricerche tra le segnalazioni già presenti nella base di dati e accedere nel caso ci si sia registrati come utente nel sito.

### 3.2.2 Login di un utente

Tramite il tasto accedi in altro a destra, è possibile aprire un form con i campi in cui inserire l'username e la password dell'utente registrato. In caso di errore nell'inserimento di dati viene lanciato un messaggio di errore e viene permesso il reinserimento dei parametri, in caso contrario in alto a destra il login avviene correttamente, il form si chiude e in alto a destra della pagina compare un messaggio di benvenuto con il nome dell'utente registrato e la possibilità di effettuare il logout. Inoltre il tasto per il login cambia e permette di modificare i dati del profilo registrati in precedenza.

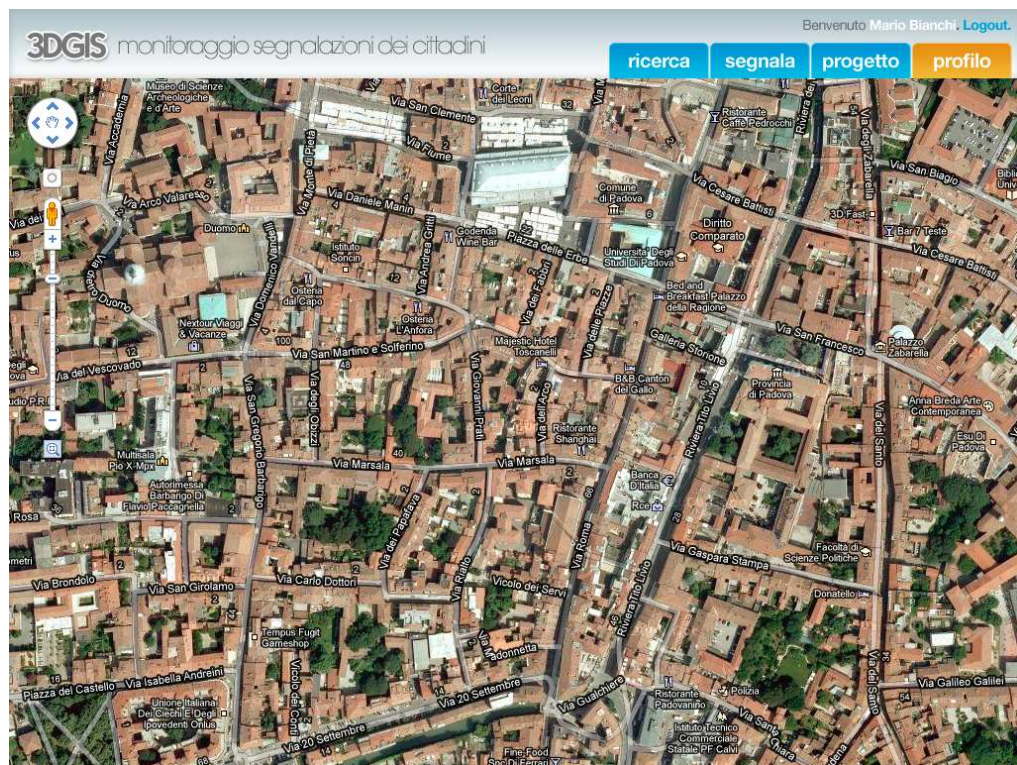


Figura 3.2: Pagina principale dopo aver effettuato il login

### 3.2.3 Inserimento di una nuova segnalazione

Partendo dalla pagina principale, cliccando sul pulsante segnala, si aprirà un dialog che chiederà all'utente se vuole effettuare il login, oppure se vuole effettuare una segnalazione senza quest'ultimo. A prescindere si aprirà un form fisso occupante la parte destra della pagina in cui sarà possibile inserire i dati geografici in cui va posizionata la registrazione, il tipo di problema, l'oggetto,

un'eventuale descrizione e foto, e i dati anagrafici nel caso l'utente abbia deciso di non effettuare il login o la registrazione.

In contemporanea all'apertura del form sulla mappa comparirà inoltre un marcatore che l'utente potrà spostare sulla mappa, in modo da registrare la posizione esatta della segnalazione; nel form inoltre comparirà la posizione del marcatore, restituendo il comune, la via e il numero civico.

Una volta inseriti tutti i parametri richiesti, avverrà la validazione: in caso di campi vuoti o errori, verrà riportato un errore e si potrà correggere o completare i campi che hanno provocato il lancio dell'errore, mentre gli altri campi rimarranno invariati con i dati inseriti dall'utente. Nel caso di un corretto inserimento, viene riportato l'avvenuto salvataggio e viene chiuso il form di inserimento.

Benvenuto Mario Bianchi. [Logout.](#)

[ricerca](#) [segnala](#) [progetto](#) [profilo](#)

**Ricerca delle segnalazioni**

Tipologia segnalazione:

Sorgente segnalazione:

Testo di ricerca:

ID	Data	Tipo	Titolo
1	10 Aprile 2009	Graffiti	Graffiti
2	23 Novembre 2010	Schiamazzi	Problema
3	01 Gennaio 2011	Dissesto stradale	Buca
4	09 Febbraio 2011	Accattonaggio	Zingari

Figura 3.3: Ricerca di una segnalazione

### 3.2.4 Ricerca di una segnalazione

Cliccando sul pulsante ricerca in alto a destra, sarà possibile cercare determinate segnalazioni in base a determinati parametri inseriti dall'utente tramite il form che comparirà sulla destra. Gli utenti potranno inserire parametri quali un particolare tipo di problema, o a un testo da immettere che andrà ricercato nei

campi delle segnalazioni presenti. Nel caso in cui non si inseriscano parametri, verranno riportate tutte le segnalazioni sulla mappa, indicate con un marcatore.

### **3.2.5 Informazioni sul servizio**

Dalla pagina principale, cliccando sul pulsante progetto è possibile visualizzare in un form tutte le informazioni riguardanti il servizio quali lo scopo del servizio, come utilizzarlo al meglio e vari link per poter, in caso di dubbi e problemi per gli utenti, comunicare direttamente con gli admin di sistema



## Capitolo 4

# Sviluppi futuri e conclusione

Nel capitolo 2 sono state enunciate tutte le funzionalità che il progetto in esame deve essere in grado di eseguire sia per utenti che per amministratori. Le funzionalità base, così come la struttura dell'applicativo, sono operative e funzionanti. Il progetto finora realizzato inoltre è già configurato per l'integrazione di futuri moduli già previsti in fase di progettazione, fra cui:

- la possibilità di effettuare e visualizzare statistiche sulla base delle segnalazioni già presentate;
- la visualizzazione ordinata dei commenti per ogni segnalazione;
- il form per gestire le comunicazioni;
- la visualizzazione dello storico dello stato per ogni segnalazione.

Tali funzionalità si renderanno disponibili a breve, in maniera tale da adempiere alle richieste dell'azienda ospitante.

Un ulteriore possibile sviluppo futuro sarà quello di rendere l'applicativo compatibile anche con le più recenti piattaforme mobile quali Android, iPhone, Blackberry e Symbian. In tal modo sarà possibile sfruttare le potenzialità dell'applicativo anche tramite il proprio smartphone, senza quindi dover per forza utilizzare un pc fisso o portatile per effettuare o ricercare segnalazioni pubbliche.





# Bibliografia

- [1] “Php.” <http://www.php.net>.
- [2] “Zend framework.” <http://framework.zend.com/>.
- [3] “Postgresql.” <http://www.postgresql.org/>.
- [4] “Postgis.” <http://postgis.refractor.net/>.
- [5] “Googlemaps.” <http://code.google.com/intl/it-IT/>.
- [6] “Jquery.” <http://jquery.com/>.
- [7] J. J. Garrett, “Ajax: A new approach to web applications.” <http://www.adaptivepath.com/ideas/e000385>.



# Elenco delle tabelle

1.1	Analisi di mercato: funzionalità degli applicativi attualmente esistenti	3
-----	--	---



# Elenco delle figure

2.1	Casi d'uso per gli utenti . . . . .	5
2.2	Casi d'uso per gli amministratori . . . . .	7
2.3	Architettura 3-tier . . . . .	9
2.4	Schema funzionamento ajax . . . . .	10
2.5	Schema logico della base di dati . . . . .	11
3.1	Pagina iniziale . . . . .	22
3.2	Pagina principale dopo aver effettuato il login . . . . .	23
3.3	Ricerca di una segnalazione . . . . .	24