

**LA TUA SPESA ONLINE:**  
**geolocalizzati e riceverai il tuo ordine**  
**comodamente a casa**

**Prof:**  
**Di Felice Marco**

**Presentato da:**  
**Margherita Capuano**  
**Grezia Di Marino**

# Indice

ABSTRACT	4
DESCRIZIONE DEL PROGETTO	7
STRUMENTI UTILIZZATI	19
GUIDA ALL'UTILIZZO	27

## Elenco delle figure

1	Strumenti utilizzati per l'area cliente. . . . .	5
2	Strumenti utilizzati per l'area commerciante. . . . .	6
3	Struttura sito Web. . . . .	7
4	Home page. . . . .	8
5	Popup per consentire la geolocalizzazione. . . . .	9
6	Area cliente: Mappa di Torricella Peligna Mappa di Torricella Peligna sud- divisa in tre zone, distinte dai differenti colori, e marker di localizzazione rappresentato dal cerchietto giallo. . . . .	10
7	Pagina Nord (a), pagina Centro (b), pagina Sud (c). . . . .	11
8	Form ordine . . . . .	12
9	Form compilato esempio ordine . . . . .	13
10	Email ricevuta esempio . . . . .	14
11	Pagina Area commerciante . . . . .	15
12	Mappa per ricercare il percorso più breve. . . . .	16
13	Popup di localizzazione per il commerciante. . . . .	17
14	Percorso più breve. . . . .	18
15	script GeoJson. . . . .	22
16	Script creazione mappa e aggiunta di tile da OSM. . . . .	25
17	Script marker. . . . .	25
18	Controllo di routing e geocodifica . . . . .	26

# ABSTRACT

A partire dal marzo 2020 stiamo vivendo un periodo di pandemia mondiale dovuta alla diffusione di un virus respiratorio: il coronavirus SARS-Cov-2. Data la velocità e le modalità di diffusione del virus, per limitarne la trasmissione, sono state adottate misure precauzionali restrittive, che hanno obbligato tutta la popolazione a restare in casa per quasi tre mesi.

In questo palcoscenico si inserisce questo lavoro, che riguarda la progettazione strutturale di un prototipo di sito Web come strumento di ausilio per ordinare e ricevere direttamente a casa generi di prima necessità.

Questo sito si suppone essere programmato e messo a disposizione dal Comune del piccolo paese abruzzese in provincia di Chieti, Torricella Peligna.

Supponiamo, quindi, che il Comune abbia progettato tale strumento per aiutare sia i cittadini a fare la spesa in tutta semplicità ricevendola comodamente a casa, sia i commercianti a ricercare il percorso più breve per effettuare la consegna dell'ordine ricevuto.

## *Scopo del progetto:*

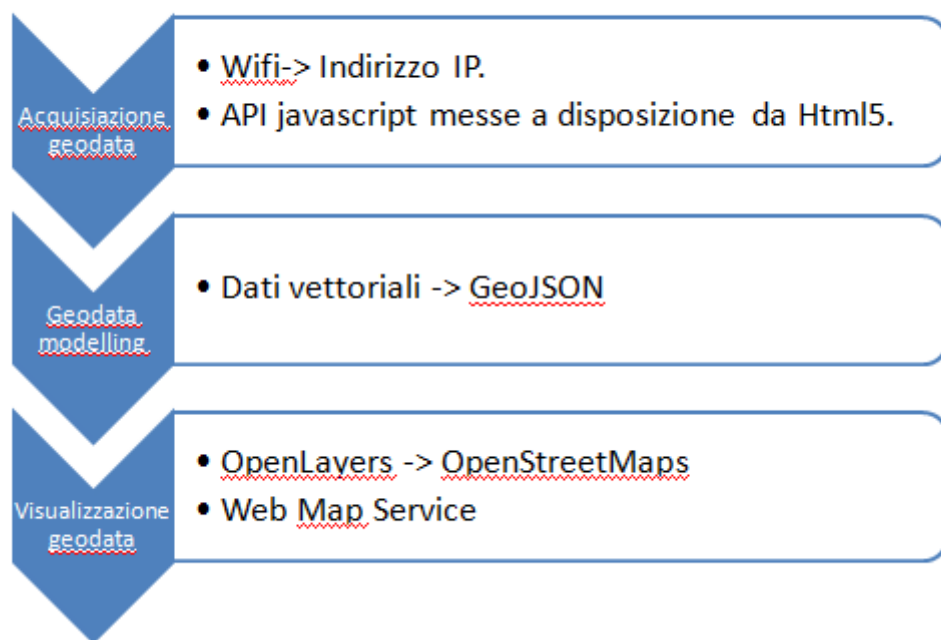
l'area territoriale del comune di Torricella Peligna è stata suddivisa da noi in tre zone: Nord, Centro e Sud. Attraverso il sito Web il cittadino può localizzarsi e accedere all'area di appartenenza (Nord-Centro-Sud).

In ognuna delle tre zone del Paese sono presenti attività commerciali di prima necessità: ortofrutta, pescheria, panetteria, farmacia, generi alimentari, macelleria, proshop. Se in un'area territoriale di appartenenza non è presente un'attività commerciale, come ad esempio una farmacia, l'utente è collegato direttamente a quella scelta di default dal sistema.

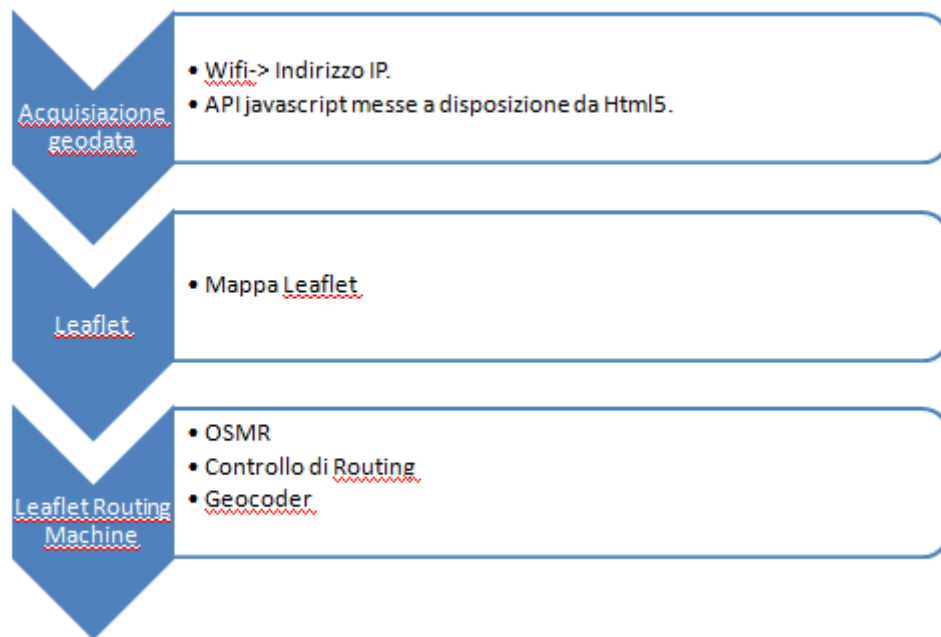
Il cittadino effettuata la localizzazione può ordinare online e ricevere comodamente a casa

la spesa. Il sito Web dispone anche di un'area riservata ai commercianti i quali possono accedere previa username e password i quali supponiamo siano stati rilasciati, in modo univoco, dal Comune (abbiamo impostato per convenzione: username “commerciante”, password “password”). Effettuato il login, il commerciante può ricercare il percorso più breve per raggiungere l'abitazione a cui consegnare l'ordine inserendo l'indirizzo ricevuto per mail, insieme alla lista della spesa.

*Strumenti utilizzati:*

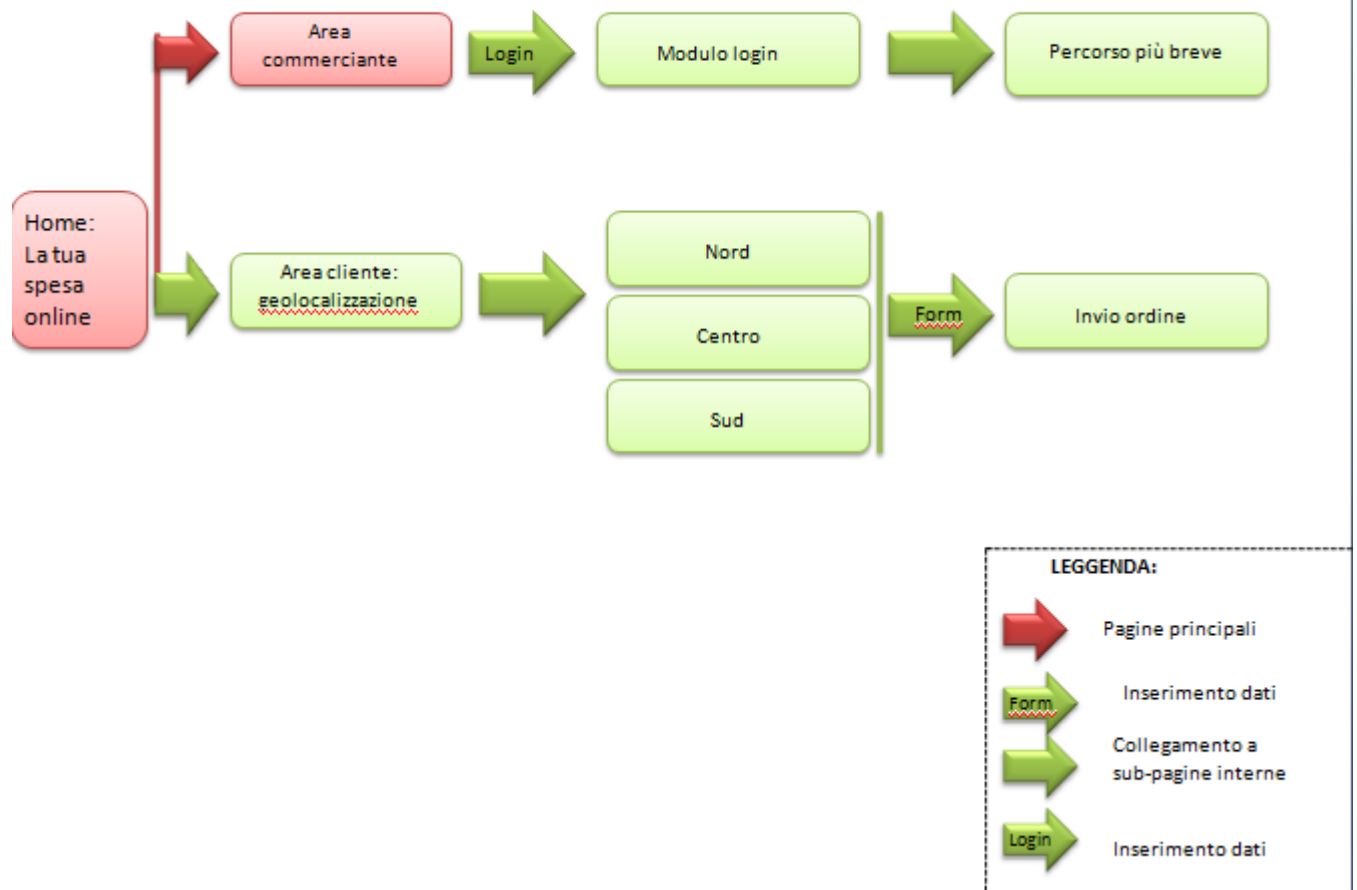


*Figura 1: Strumenti utilizzati per l'area cliente.*



*Figura 2: Strumenti utilizzati per l'area commerciante.*

## DESCRIZIONE DEL PROGETTO



*Figura 3: Struttura sito Web.*

Nel sito Web le informazioni sono strutturate in sezioni, accessibili con poche e semplici azioni. La Figura 3 rappresenta il blueprint dell'architettura dell'informazione, uno schema in cui viene illustrata una mappa concettuale delle pagine e delle funzioni presenti nel sito.



*Figura 4: Home page.*

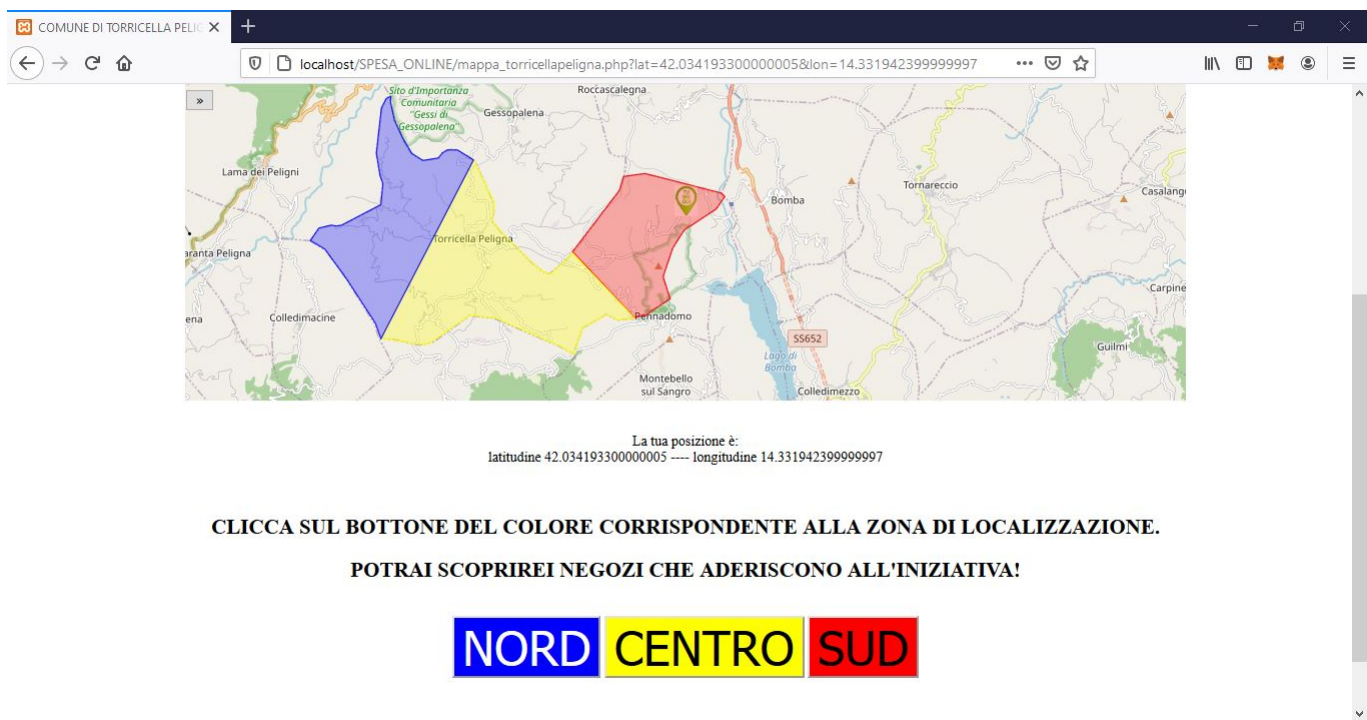
La pagina di Home è intitolata “La tua spesa online”. Da questa pagina (Figura 4), l’utente cliccando sul bottone “CLICCA QUI PER EFFETTUARE IL TUO ORDINE” potrà localizzarsi. Di fatti, gli uscirà un popup per geolocalizzarsi (Figura 5).





Figura 5: Popup per consentire la geolocalizzazione.

Se la risposta è “consento” si aprirà automaticamente la sub-pagina “Area cliente: geolocalizzazione”(Figura 18).



*Figura 6: Area cliente: Mappa di Torricella Peligna Mappa di Torricella Peligna suddivisa in tre zone, distinte dai differenti colori, e marker di localizzazione rappresentato dal cerchietto giallo.*

Qui vedrà la mappa del Paese e la sua localizzazione: la mappa è suddivisa in tre zone dai colori blu, giallo e rosso rispettivamente ad indicare nord, centro e sud. Un marker individuerà la sua posizione (localizzazione) in quell'area. Conosciuta la sua area di appartenenza, gli basterà cliccare il bottone “NORD”, o “CENTRO”, o “SUD” e così sarà indirizzato alle relative sub-pagine, intitolate rispettivamente “Nord” (Figura 7a), “Centro” (Figura 7b) e “Sud”(Figura 7c), per scoprire i negozi aderenti a questa iniziativa.

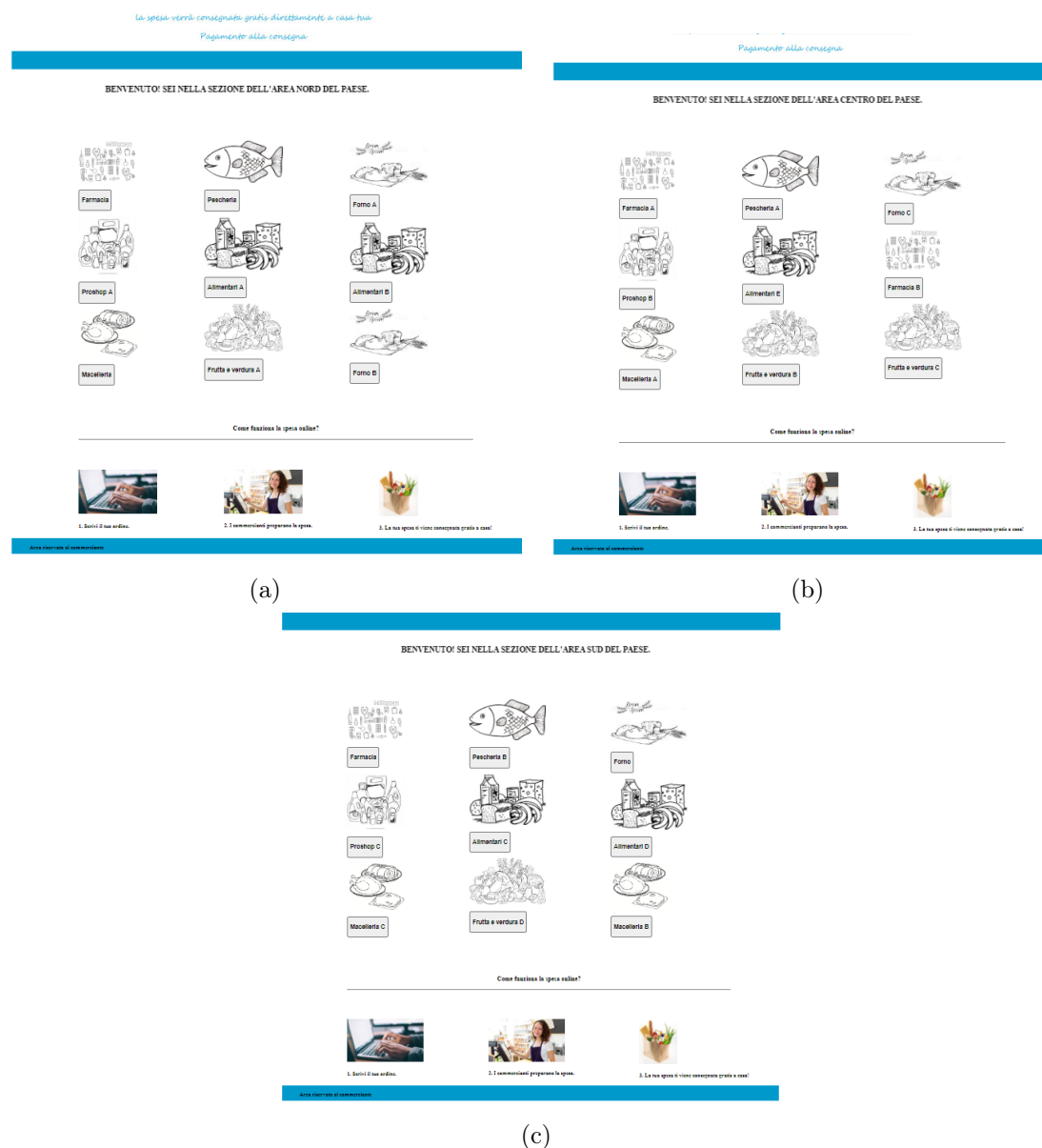
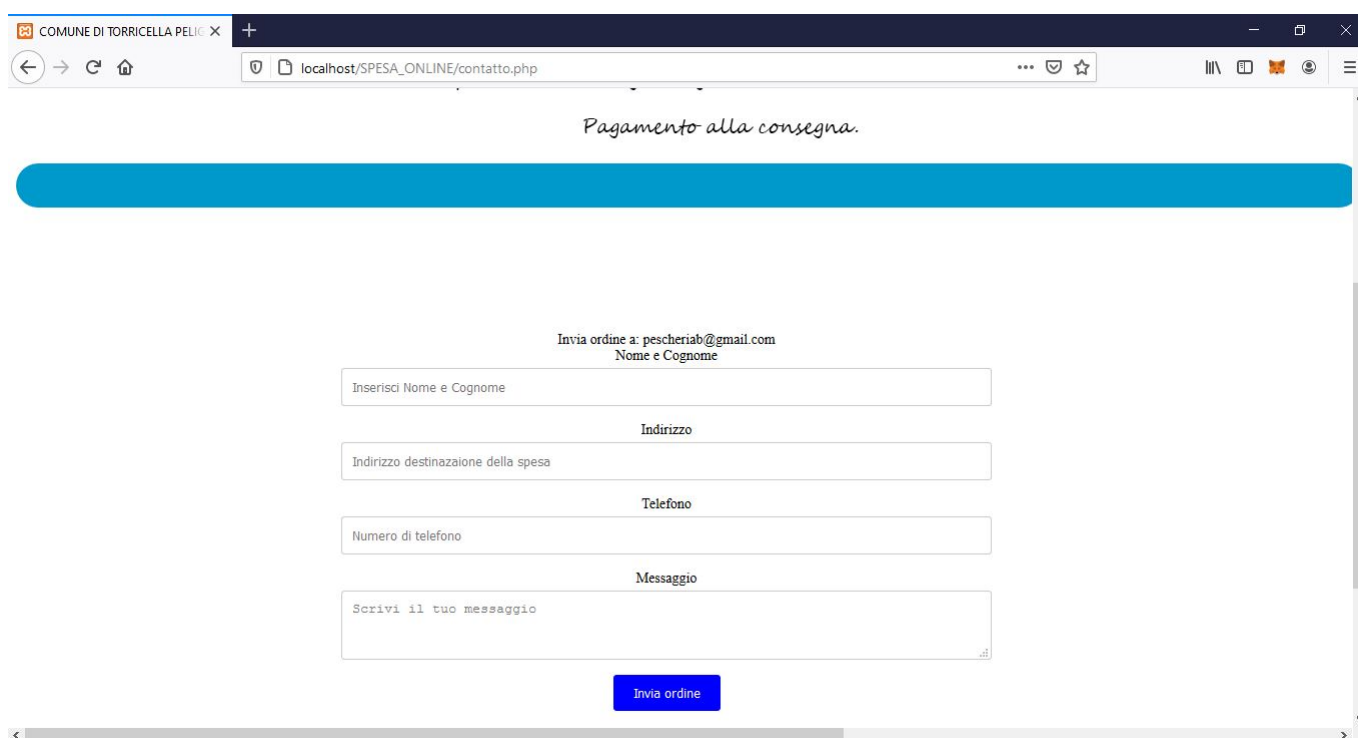


Figura 7: Pagina Nord (a), pagina Centro (b), pagina Sud (c).

I negozi presenti sono attività commerciali di prima necessità: ortofrutta, pescheria, panetteria, farmacia, generi alimentari, macelleria, proshop. Se in un'area territoriale di appartenenza non è presente un'attività commerciale, come ad esempio una farmacia, l'utente è collegato direttamente a quella scelta di default dal sistema. In queste sub-

pagine, l'utente potrà selezionare il negozio a cui vuole effettuare l'ordine cliccando sul bottone riportante il nome dell'attività ( per convenzione abbiamo attribuito ai negozi nomi del tipo "Forno A, Proshop A, Macelleria B" e così via), così facendo accederà alla sub-pagina "Invio ordine" dove troverà il form (Figura 8) da compilare in ogni suo campo per spedire, direttamente all'attività selezionata, l'ordine desiderato (nella Figura (9 si riporta un'esempio).



The screenshot shows a web browser window with the address bar displaying 'localhost/SPESA\_ONLINE/contatto.php'. The page title is 'Pagamento alla consegna.' and it features a blue header bar. The form is titled 'Invia ordine a: pescheriab@gmail.com' and 'Nome e Cognome'. It contains four input fields: 'Inserisci Nome e Cognome', 'Indirizzo' (with subtext 'Indirizzo destinazione della spesa'), 'Telefono' (with subtext 'Numero di telefono'), and 'Messaggio' (with subtext 'Scrivi il tuo messaggio'). A blue 'Invia ordine' button is at the bottom.

COMUNE DI TORRICELLA PELIC X

localhost/SPESA\_ONLINE/contatto.php

Pagamento alla consegna.

Invia ordine a: pescheriab@gmail.com  
Nome e Cognome

Inserisci Nome e Cognome

Indirizzo

Indirizzo destinazione della spesa

Telefono

Numero di telefono

Messaggio

Scrivi il tuo messaggio

Invia ordine

Figura 8: Form ordine

COMUNE DI TORRICELLA PELICIA X

localhost/SPESA\_ONLINE/contatto.php

Pagamento alla consegna.

Invia ordine a: pescheriab@gmail.com  
Nome e Cognome

grezia di marino

Indirizzo

bufalara 7

Telefono

3298-----

Messaggio

50gr gamberi  
50gr cozze  
50gr vongole

Invia ordine

*Figura 9: Form compilato esempio ordine*

Inviato il messaggio, il commerciante riceve una mail contenente del mittente il nome e cognome , indirizzo a cui effettuare la consegna, il recapito telefonico e lista della spesa (nella Figura 10) si riporta un'esempio).

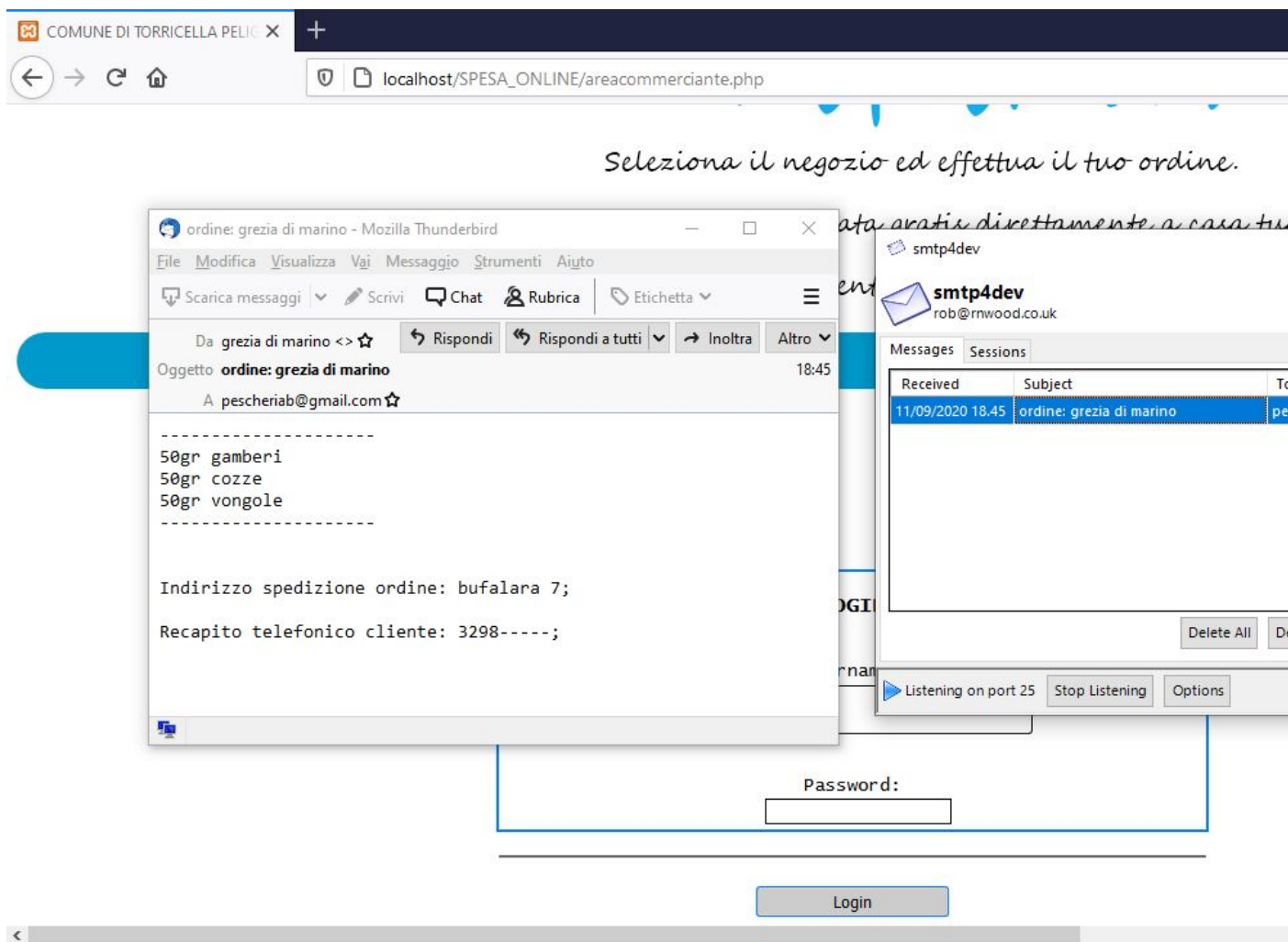
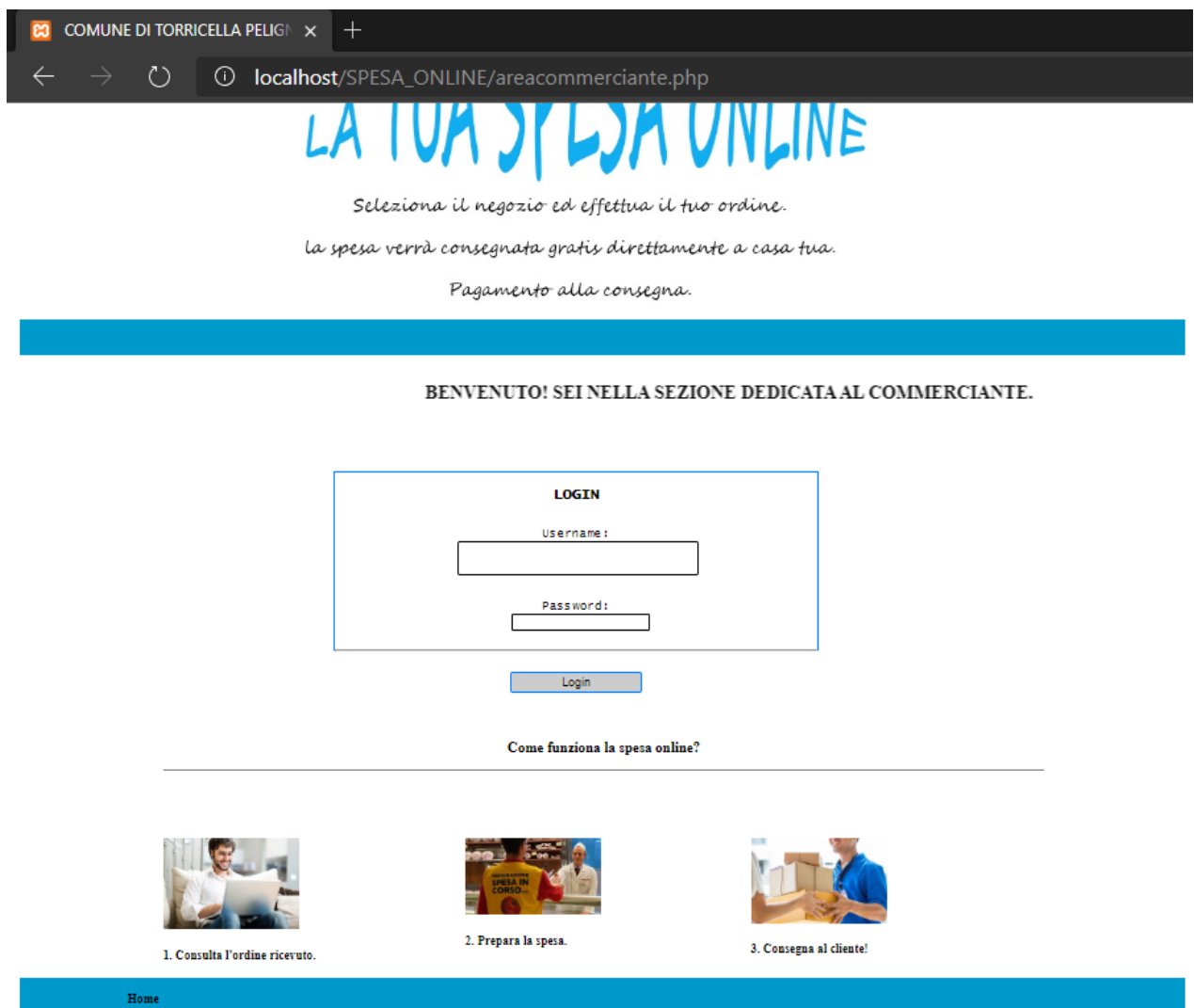


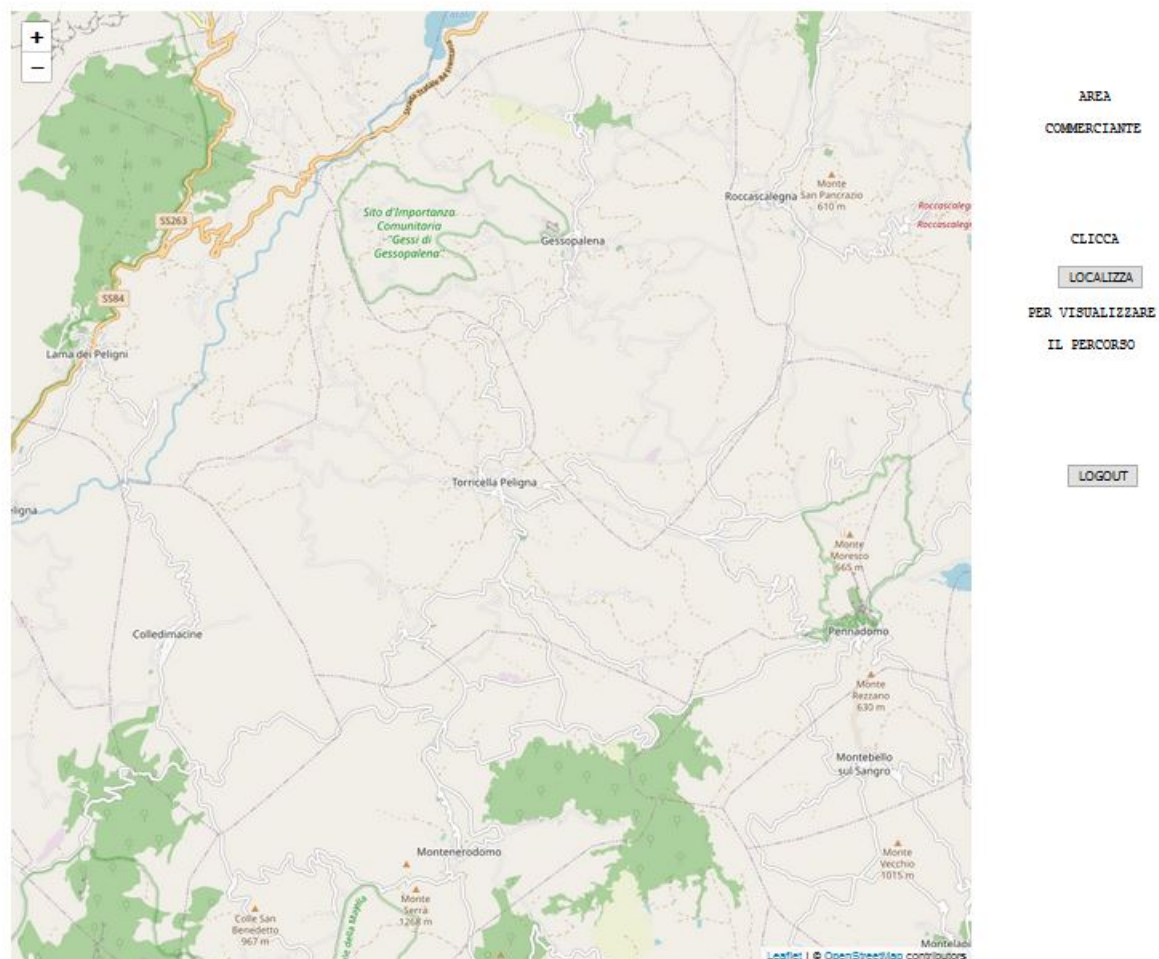
Figura 10: Email ricevuta esempio

Nella Home page, nel footer menu è presente un bottone “Area riservata al commerciante”, cliccandoci si accede alla seconda pagina principale del sito Web, quella dedicata al commerciante (Figura 11) .



*Figura 11: Pagina Area commerciante*

Da questa pagina, intitolata “Modulo di login”, il commerciante dovrà compilare il modulo di login con username e password (supponiamo che queste credenziali siano state rilasciate dal Comune ad ognuno di essi e che siano: username: “commerciantephp”, password: “password”) per essere indirizzato alla sub-pagina “Percorso”(Figura 12).



*Figura 12: Mappa per ricercare il percorso più breve.*

Da questa sub-pagina il commerciante può ricercare il percorso più breve per raggiungere l'abitazione a cui consegnare la spesa. Cliccando sul pulsante "Clicca" gli uscirà un popup per consentire o meno di rilevare la sua posizione (Figura 13).



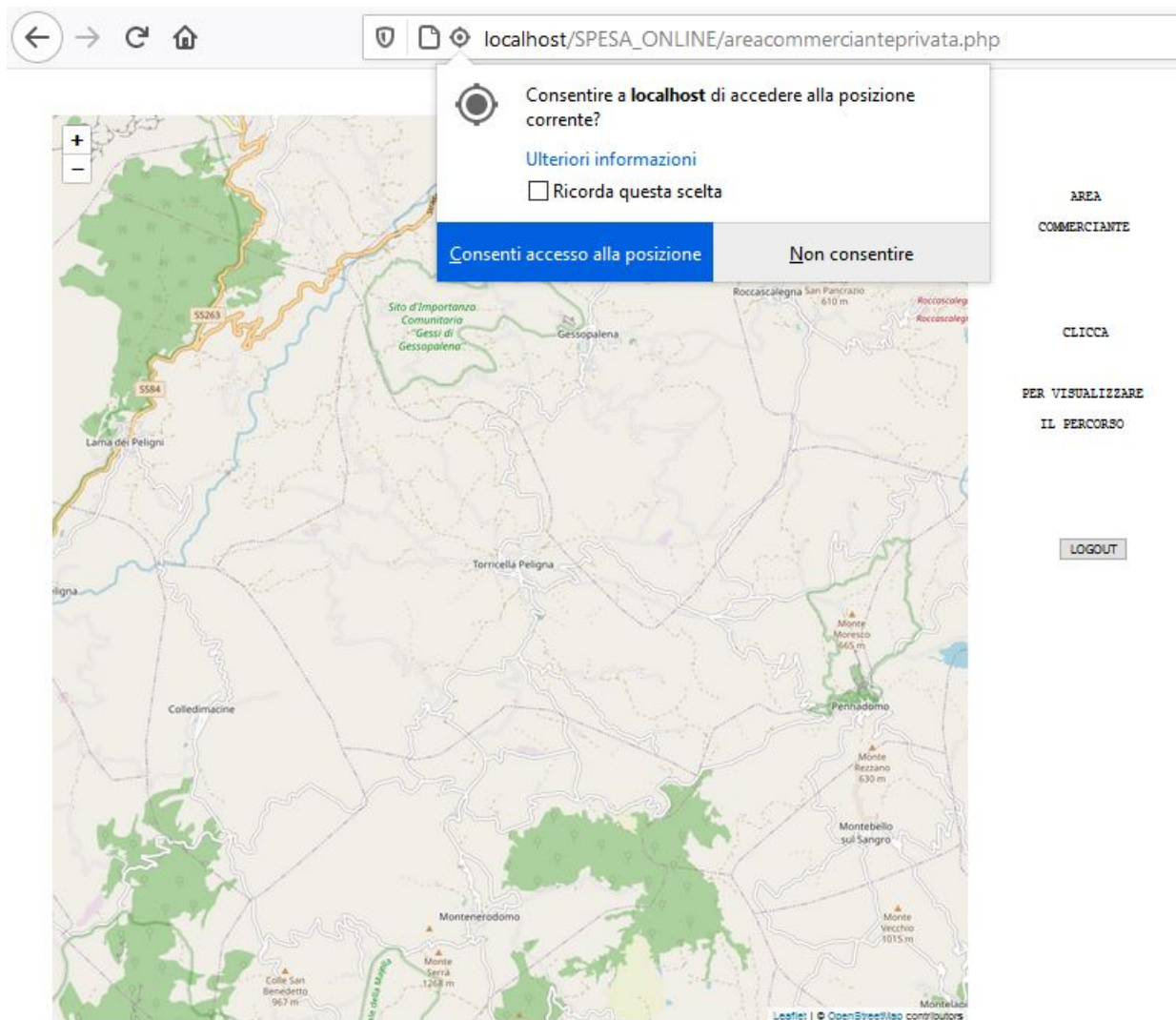


Figura 13: Popup di localizzazione per il commerciante.

Se la risposta è "consento", si aprirà un box dove è possibile inserire da tastiera l'indirizzo a cui consegnare la spesa (indirizzo è ricevuto tramite mail). Il percorso è tracciato con il colore rosso e nel box è possibile leggere anche le indicazioni stradali per raggiungere l'abitazione. Un esempio è riportato nella Figura 14.

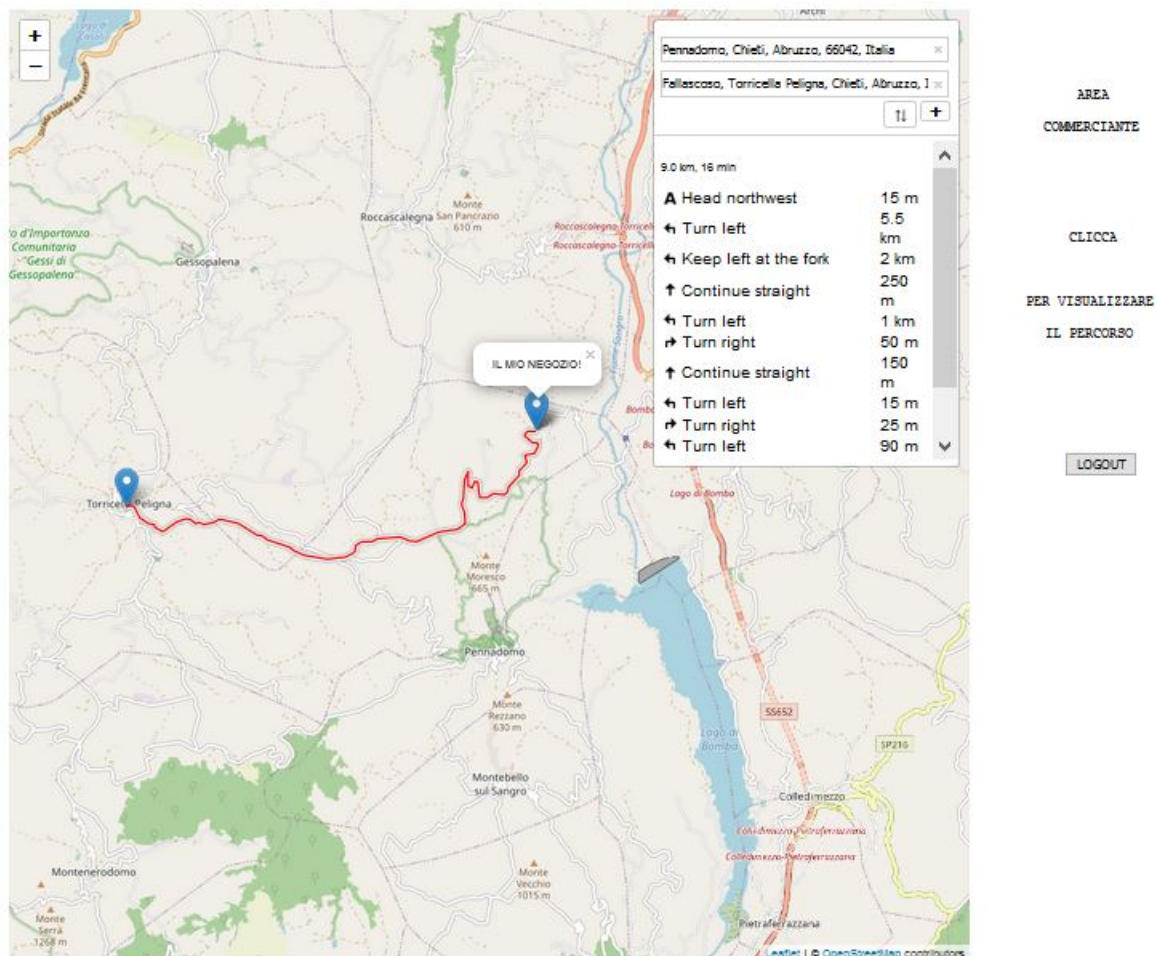


Figura 14: Percorso più breve.

I linguaggi di programmazione utilizzati per realizzare il sito Web sono html, css, php e javascript. Ogni pagina, del sito Web realizzato, è un documento con estensione .php, e tutti i documenti sono salvati in una stessa directory. Tutti i documenti utilizzano lo standard html5, e il collegamento al foglio di stile esterno avviene mediante la direttiva import.

# STRUMENTI UTILIZZATI

## Geolocalizzazione

La geolocalizzazione è l' identificazione della posizione geografica di un cellulare o di un computer o di qualsiasi altro oggetto connesso o meno ad Internet . L'identificazione può avvenire secondo diverse tecniche.

La tecnica da noi utilizzata, sia per l'area cliente che per quella del commerciante, è quella basata sulla rete WIFI, cioè sul segnale della fonte WIFI, la quale a sua volta è localizzata tramite la rete Internet ovvero tramite l'indirizzo IP della propria rete internet dove la localizzazione dipende dalla registrazione del collegamento ad una base di dati.

Di seguito riportiamo nel dettaglio come avviene quanto detto nelle righe precedenti.

Il browser da noi utilizzato per lanciare e testare il sito Web creato è Firefox. All'utente viene chiesto se è possibile accedere alla propria posizione mediante una finestra popup e se si permette l'accesso verranno raccolte le informazioni sui punti di accesso wireless nelle vicinanze e sull'indirizzo IP del computer dell'utente. Successivamente questi dati verranno trasmessi da Firefox al fornitore predefinito per il servizio di georilevazione, ovvero Google Location Services, per ricevere una stima della posizione dell'utente (latitudine e longitudine). Infine questa informazione verrà comunicata al sito Web che ne ha fatto richiesta.

Firefox per determinare la posizione dell'utente invia i seguenti dati a Google Location Services:

- L'indirizzo IP del computer dell'utente.
- Informazioni sui punti di accesso wireless nelle vicinanze.
- Un codice identificativo casuale del client, assegnato da Google, che scade ogni due settimane.

Per garantire la privacy dell'utente lo scambio di informazioni avviene utilizzando una connessione cifrata e il nome del sito che si sta visitando o i cookie impostati da questo sito non vengono mai trasmessi a Google Location Services.

Ciò che fa scattare la richiesta di autorizzazione a rilevare la posizione è l'utilizzo delle API javascript per la geolocalizzazione messe a disposizione nativamente da HTML5. Infatti nel file `mappa_torricellapeligna.php` viene aggiunto uno script dove si verifica:

1. innanzitutto se il browser supporti o meno la geolocalizzazione.
2. Se la geolocalizzazione è supportata, si esegue il metodo `getCurrentPosition()`,
3. in caso contrario si visualizza un messaggio di errore.
4. Attraverso il metodo `getCurrentPosition()` recuperiamo i dati sulla posizione del dispositivo. Questo metodo passa due argomenti: la funzione `showPosition` per restituire le coordinate longitudine e latitudine e la funzione `showError` da eseguire se non si riesce ad ottenere la posizione.
5. Nella funzione `showError` disponiamo quattro codici di errori principali: l'utente ha rifiutato la richiesta di geolocalizzazione, le informazioni sulla posizione non sono disponibili, la richiesta di ottenere la posizione dell'utente è scaduta, si è verificato un errore sconosciuto.

## **Strumenti utilizzati per area cliente**

I dati spaziali da noi utilizzati sono dati vettoriali cioè le caratteristiche geografiche sono rappresentate come geometrie discrete, nel nostro caso come poligono ovvero come un' area chiusa. I dati spaziali, memorizzano sia la posizione geografica di particolari caratteristiche che le informazioni che descrivono cosa rappresentano queste caratteristiche.

Come modello di dati spaziali vettoriali utilizziamo GeoJson.

GeoJson è un formato per la codifica di una varietà di strutture di dati geografici, basato su JSON (JavaScript Object Notation), ed è stato rilasciato nel 2005 dalla Internet Engineering Task Force. Un file JSON è anche chiamato "Documento" ed è supportato da una varietà di piattaforme GIS e da API per la visualizzazione di mappe come ad es. Google Maps v3 ed OpenLayers.

Un documento JSON è racchiuso tra parentesi ed ogni dato inserito è una coppia <chiave, valore>. Nel nostro lavoro un oggetto GeoJson ha tipo "FeatureCollection" ed è una raccolta di caratteristiche. Un oggetto di questo tipo ha un membro "features" il cui valore corrispondente è un array dove, a sua volta, ogni suo elemento è un feature object. In pratica attraverso GeoJson creiamo i layers vettoriali delle tre zone Nord, Centro, Sud del paese che poi andremo ad aggiungere alla nostra mappa. Di seguito riportiamo il codice GeoJson (Figura 15) riferito ad una sola zona geografica, quella del Nord, come esempio; le altre sono state programmate nel medesimo modo.

```

1  {
2    "type": "FeatureCollection",
3    "features": [
4      {
5        "type": "Feature",
6        "properties": {
7
8        },
9        "geometry": {
10         "type": "Polygon",
11         "coordinates": [
12           [
13             [
14               14.314584732055662,
15               42.00427965240744
16             ],
17             [
18               14.318275451660156,
19               42.00491745263065
20             ],
21           ]
22         ],
23         ....
24         ....
25       ]
26     ],
27     [
28       14.293212890625,
29       42.0211154345676
30     ],
31     [
32       14.314584732055662,
33       42.00427965240744
34     ]
35   ]
36 }
37 }
38 ]
39 }

```

*Figura 15: script GeoJson.*

Per inserire nell'area cliente una mappa abbiamo utilizzato OpenLayers, una libreria Javascript il quale supporta il formato vettoriale dei dati geografici di GeoJson e tra i differenti rendering della mappa vi è anche HTML5, quello da noi scelto.

OpenLayers è un software completamente gratuito, JavaScript Open Source, rilasciato sotto la licenza BSD.

La versione che utilizziamo è la 6.3.1.

Per poter lavorare con OpenLayers abbiamo scaricato il package ol e per poter accedere alle fonti di informazioni cartografiche in Internet, il software offre API come ad esempio OpenStreetMap, Google Maps, Bing ecc. L' API che abbiamo studiato per il nostro lavoro è OpenStreetMap (OSM). Il package ol include una build della libreria ol.js, una mappa sorgente ol.js.map con informazioni sulla struttura e source della mappa e una libreria CSS ol.css con mappa sorgente ol.css.map.

Abbiamo creato una mappa elementare di Torricella Peligna e salvato il lavoro nella directory SPESA\_ONLINE (SPESA\_ONLINE\mappa\_torricellapeligna.php).

La mappa è divisa in una sequenza di celle quadrate (tiles) con dimensione 256x256 pixel.

La mappa è rappresentata dalla classe ol.Map ed è stato creato un layer di tile dove la mappa mostra i contenuti, rappresentato dalla classe ol.layer.Tile che a sua volta carica il contenuto dal progetto OpenStreetMap usando la classe sorgente ol.source.OSM. E' stata inserita la classe ol.View che determina la posizione iniziale della "camera" e come la mappa è stata reindirizzata (zoom iniziale , posizione centrale iniziale ). Abbiamo aggiunto a questa mappa i layers vettoriali rappresentati da ol.layer.Vector con source GeoJson file; nelle specifico sono tre layers corrispondenti alle tre zone che prendiamo in considerazione:

- SPESA\_ONLINE\ geojson\ zona1.geojson
- SPESA\_ONLINE\geojson\zona2.geojson
- SPESA\_ONLINE\geojson\zona3.geojson

Inoltre abbiamo aggiunto a questi layers uno stile identificativo.

Abbiamo inserito anche un marker il quale evidenzia con un simbolo la posizione dove si trova l'oggetto localizzato, nel nostro caso indica la posizione dell'utente che ha effettuato l'ordine .

Per poter accedere alle fonti di informazioni cartografiche in Internet abbiamo utilizzato l'API OpenStreetMap (OSM): un progetto fondato nel 2004 da Steve Coast per permette di creare mappe.

I dati geografici presenti in OSM sono distribuiti sotto licenza libera, la Open Database License. Le mappe sono create usando dati registrati da varie sorgenti come ad esempio tracce GPS, fotografie aeree ed altre fonti libere. Tutti posso contribuire ad aggiungere e modificare le mappe (crowdsourcing).

Per servirci di immagini di mappe georeferenziate generate da un server di mappe facciamo uso dello Standard WMS dell' Open GeoSpatial Consortium (OGC). L'acronimo WMS sta letteralmente per Web Map Service. Le immagine della mappa prodotta da WMS hanno formato png e le operazioni vengono invocate usando `getRequsestUrl()`, richiedendo l'URL utilizzato dal client per effettuare una richiesta.

Nel richiedere una mappa la URL indica quali informazioni devono essere visualizzate sulla mappa, quale porzione della Terra deve essere rappresentata, il sistema di coordinate desiderato, il formato e le dimensioni dell'immagine di output.

Tutte le informazioni su WMS sono riportate nei file del package ol.

## **Strumenti utilizzati per area commerciante**

La mappa che si trova nell'area commerciante è stata creata utilizzando la libreria Leaflet. Si tratta di una libreria javascript open source per mappe interattive sviluppata nel 2010 che supporta la maggior parte dei browser e degli standard HTML5 e CSS3.

Leaflet permette di mostrare punti di interesse, livelli ecc su una mappa a tasselli.



```
//abbiamo creato una mappa nel 'map' div, impostando le coordinate e lo zoom
var map = L.map('map').setView([42.0239700,14.2585400],12);

//aggiunta delle mappa a tasselli da openstreetmap
L.tileLayer('https://{s}.tile.osm.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="https://osm.org/copyright">OpenStreetMap</a> contributor
}).addTo(map);
```

*Figura 16: Script creazione mappa e aggiunta di tile da OSM.*

```
//marker con coordinate del commerciante restituite dalla funzione getLocation()
L.marker([lat, lon]).addTo(map)
  .bindPopup('IL MIO NEGOZIO!')
  .openPopup();
}
```

*Figura 17: Script marker.*

Il nostro lavoro è consistito: nel creare una mappa (Figura 16), aggiungere tile da OpenStreetMap e inserire un marker (Figura 17).

Alla mappa creata, abbiamo poi aggiunto un percorso più breve esistente tra due punti con Leaflet Routing Machine: la versione da noi utilizzata è la 3.2.12. Leaflet Routing Machine è un modo (API) di routing open source, rilasciato sotto licenza ISC ed è supportato da diversi engine routing: vi è uno predefinito, Open Source Routing Machine (OSRM).

OSRM è un'implementazione C++ di un motore di routing ad alte prestazioni per i percorsi più brevi nelle reti stradali. Con licenza in base alla licenza BSD a 2 clausole, è un servizio di rete gratuito che Combina algoritmi di calcolo del percorso con i dati della rete stradale aperta e gratuita del progetto OpenStreetMap (OSM). Il Routing è dinamico "click-to-drag", come Google Maps, e permette percorsi alternativi.

```
// controllo di routing (box) --> aggiungiamo alla mappa un controllo di routing con waypoints
var control = L.Routing.control({

  waypoints: [
    L.latLng(lat,lon), //latitudine e longitudine del commerciante
    L.latLng() // inserire indirizzo da tastiera del cliente
  ],

  //processo di geocodifica utilizzando nominatim per rispondere alla query di geocodifica
  geocoder: L.Control.Geocoder.nominatim(),
  routeWhileDragging: true,
  reverseWaypoints: true,
  showAlternatives: true,
  altLineOptions: {
    styles: [
      {color: 'yellow', opacity: 0.15, weight: 9},
      {color: 'red', opacity: 0.8, weight: 6},
      {color: 'green', opacity: 0.5, weight: 2}
    ]
  }
}).addTo(map);
```

*Figura 18: Controllo di routing e geocodifica*

Per poter lavorare con Leaflet Routing Machine nella pagina `areacommerciantepri-  
ta.php` abbiamo incluso `leaflet-routing-machine.css` e `leaflet-routing-machine.js`.

Abbiamo aggiunto il controllo di routing utilizzando il codice Javascript riportato nella Figura 18.

Poiché il software di instradamento può solo instradare tra posizioni, latitudini e longitu-  
dini, il software necessita di un modo per cercare le coordinate di un indirizzo: abbiamo  
quindi aggiunto un processo di geocodifica, ovvero di ricerca di latitudine e longitudine  
da una stringa di indirizzo (Figura 18). Gli scripts di creazione della mappa e controllo  
di routing sono presenti nel file `index.js`.

# GUIDA ALL'UTILIZZO

Si riportano delle linee guida, illustrando i passaggi per aprire il sito web creato.

- Scaricare la cartella "SPESA\_ONLINE" e decomprimerla.
- Scaricare un application server capace di interpretare pagine web dinamiche, noi abbiamo utilizzato XAMPP.
- Scaricare un server di posta elettronica SMTP, noi abbiamo utilizzato smtp4dev.
- Aprire il browser e inserire l'url del progetto aprendo la homepage (inizio.php), es localhost/SPESA\_ONLINE/inizio.php. Il browser su cui sono state fatte le prove in fase di programmazione è Firefox.