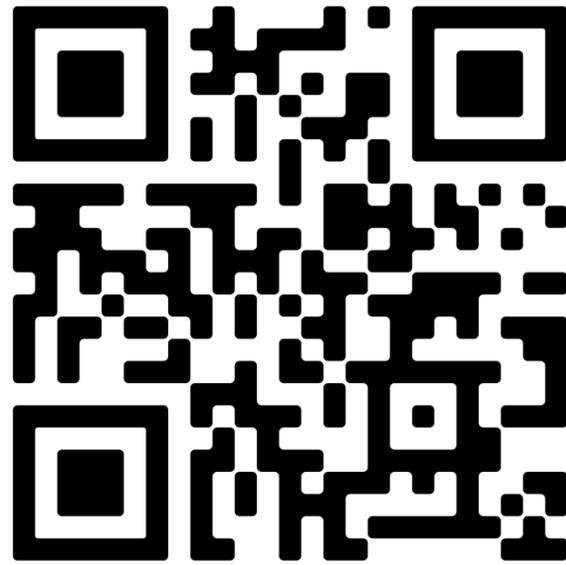


TecnoTerr

tecnologia e territorio



PRIMA DI INCOMINCIARE



Scannerizzate il QR per scaricare
l'applicazione che servirà in seguito



NORMATIVA LOMBARDIA

«Le imprese sono obbligate a giustificare il trasporto di effluente all'interno della stessa azienda mediante il "Registro delle distribuzioni di fertilizzanti" Per distanze superiori a 40 Km in linea d'aria, il trasporto deve essere giustificato da apposita registrazione, effettuata mediante sistemi di posizionamento geografico (GPS); copia della registrazione deve essere conservata presso l'azienda, a disposizione per eventuali controlli, per almeno due anni...»

SFIDA

Creazione di un applicazione che consenta di tracciare gli spostamenti dei mezzi agricoli trasportanti nitrati (tramite GPS) in grado di collegarsi ad un database per recuperare informazioni cartografiche, verificare che il trasporto avvenga presso il poligono corretto e di poter esportare i dati raccolti

DATI CARTOGRAFICI

I dati cartografici sono informazioni geografiche dettagliate che rappresentano il mondo reale, inclusi punti di interesse, strade, confini, altitudini e altro ancora.

Alcuni formati:

- **Shapefile** (.shp)
- **KML** (Keyhole Markup Language) (.kml, .kmz)
- **GPKG** (GeoPackage) (.gpkg)
- **WKT** (Well-Known Text) (.wkt) e **WKB** (Well-Known Binary) (.wkb)

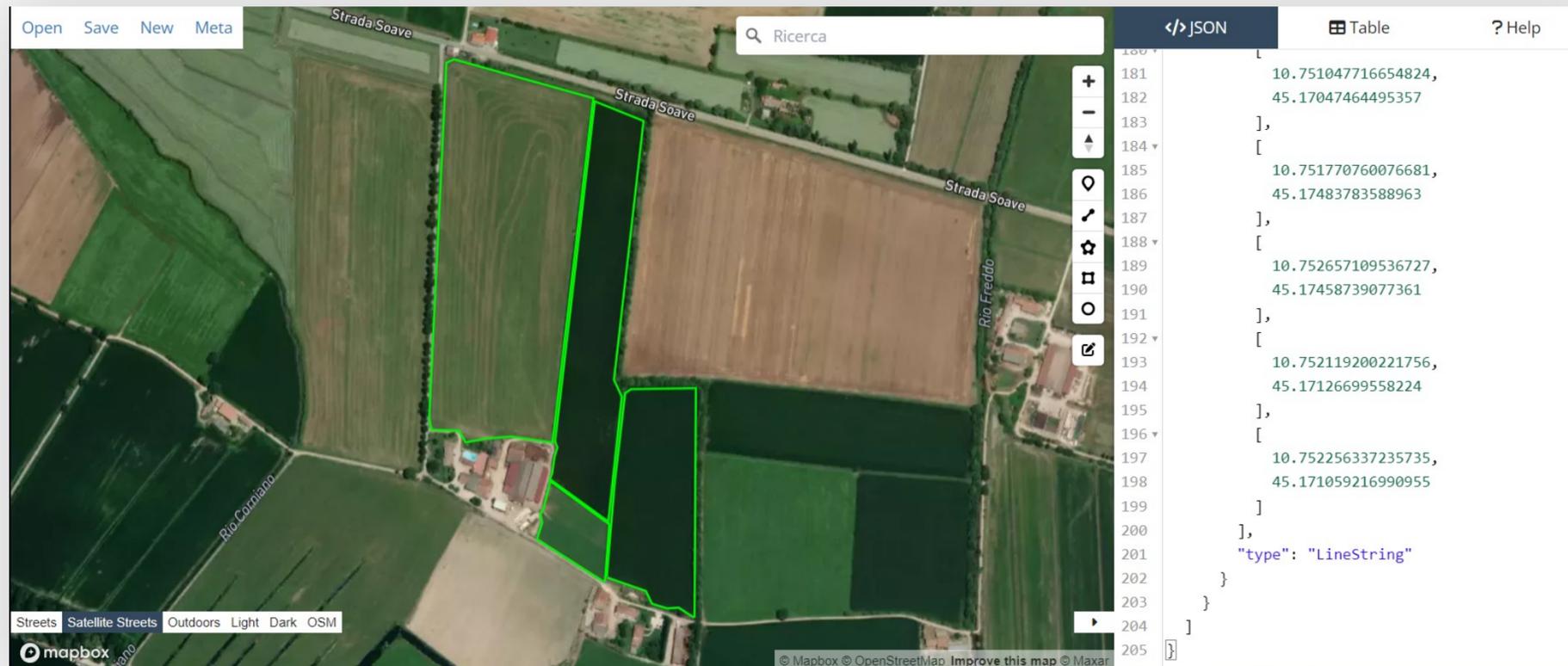


Esempio di dato cartografico

GeoJSON è uno standard aperto utilizzato per lo scambio di dati geospaziali che descrive feature geografiche, inclusi i loro attributi non spaziali. Il formato si basa su **JavaScript Object Notation (JSON)** e consente la codifica di varie strutture di dati geografiche.

Supporta diversi tipi di feature, tra cui:

- **Punto** (inclusi indirizzi e posizioni)
- **Stringa lineare** (incluse strade, autostrade e confini)
- **Poligono** (inclusi paesi, province e sezioni di terreni)



DATABASE

Esempio di creazione di un Database semplificato tramite MySQL contenente i dati relativi l'azienda

ID_Azienda	NomeAzienda	CodiceIdentificativo
1	Azienda1	AZ001

Tabella Azienda

id	coordinate_geo	data_ora	velocita	direzione
1	0x0000000001010000005B3FFD67CD8F4640309B00C3F2271F40	2023-09-28 13:45:00	20.50	120.00
2	0x0000000001010000009694BBCFF18F4640C8940F41D5281F40	2023-09-28 13:50:00	18.30	112.50
3	0x000000000101000000D1E9793716904640C2BF081A33291F40	2023-09-28 13:55:00	22.10	128.90

Tabella spostamento mezzo agricolo

ID_Profilo	ID_Azienda	GeoJSON	CodiceIdentificativoPoligono	ID_Tipologia_Nitrati
1	1	NULL	PT001	NULL
2	1	NULL	PT001	NULL

Tabella Poligoni Azienda

ID_Tipologia	NomeTipologia
1	Nitrate1

Tabella Tipologia Nitrati

ID_Sversamento	ID_Profilo	ID_Tipologia_Nitrati	Data	Ora	QuantitaSversata	KmPercorsi
1	1	1	2023-09-25	14:30:00	50.00	10.00

Tabella Sversamento Nitrati



TRACCIAMENTO PERCORSI

Il GPS viene collegato alla batteria del trattore mentre sulla cisterna viene applicato un GPS con batteria dalla durata di circa 5/7 anni.



GPS tracker



Esempio di geolocalizzazione tramite GPS Tracker



COMUNICAZIONE TRA TRACKER E APP

Questo può essere fatto utilizzando un linguaggio di programmazione come Python, Java, o qualsiasi linguaggio compatibile con il database.

Dopodiché si avvia il programma per il trasferimento dei dati.

Una volta che i dati sono archiviati nel database, si può utilizzare query SQL o strumenti di analisi per estrarre informazioni utili dai dati di spostamento, come tracciare le rotte del trattore e calcolare le distanze percorse

Si utilizza la libreria **Shapely** per gestire i poligoni e calcolare la distanze tra la posizione GPS corrente e il punto più vicino nel poligono dell'area.



```
1 import gpsd
2 from shapely.geometry import Point, Polygon
3 from shapely.ops import nearest_points
4
5 # Definizione del poligono dell'area predefinita
6 area_polygon = Polygon([(45.0, -75.0), (45.0, -76.0), (46.0, -76.0), (46.0, -75.0)])
7
8 # Distanza massima consentita dall'area in metri
9 distanza_massima = 100.0 # valore regolabile in base alle esigenze
10
11 # Connessione a gpsd
12 gpsd.connect()
13
14 try:
15     while True:
16         packet = gpsd.get_current()
17         if packet.mode == 3:
18             # Crea un oggetto Point con le coordinate GPS correnti
19             current_location = Point(packet.lat, packet.lon)
20
21             # Calcolo del punto più vicino nel poligono
22             nearest_point = nearest_points(area_polygon, current_location)[0]
23
24             # Calcolo della distanza tra il punto corrente e il punto più vicino del poligono
25             distance = current_location.distance(nearest_point)
26
27             if distance <= distanza_massima:
28                 print("Hai raggiunto l'area predefinita!")
29                 break
30     except KeyboardInterrupt:
31         pass
32     finally:
33         gpsd.close()
```

Esempio script in Python per raggiungimento poligono predefinito

USER EXPERIENCE

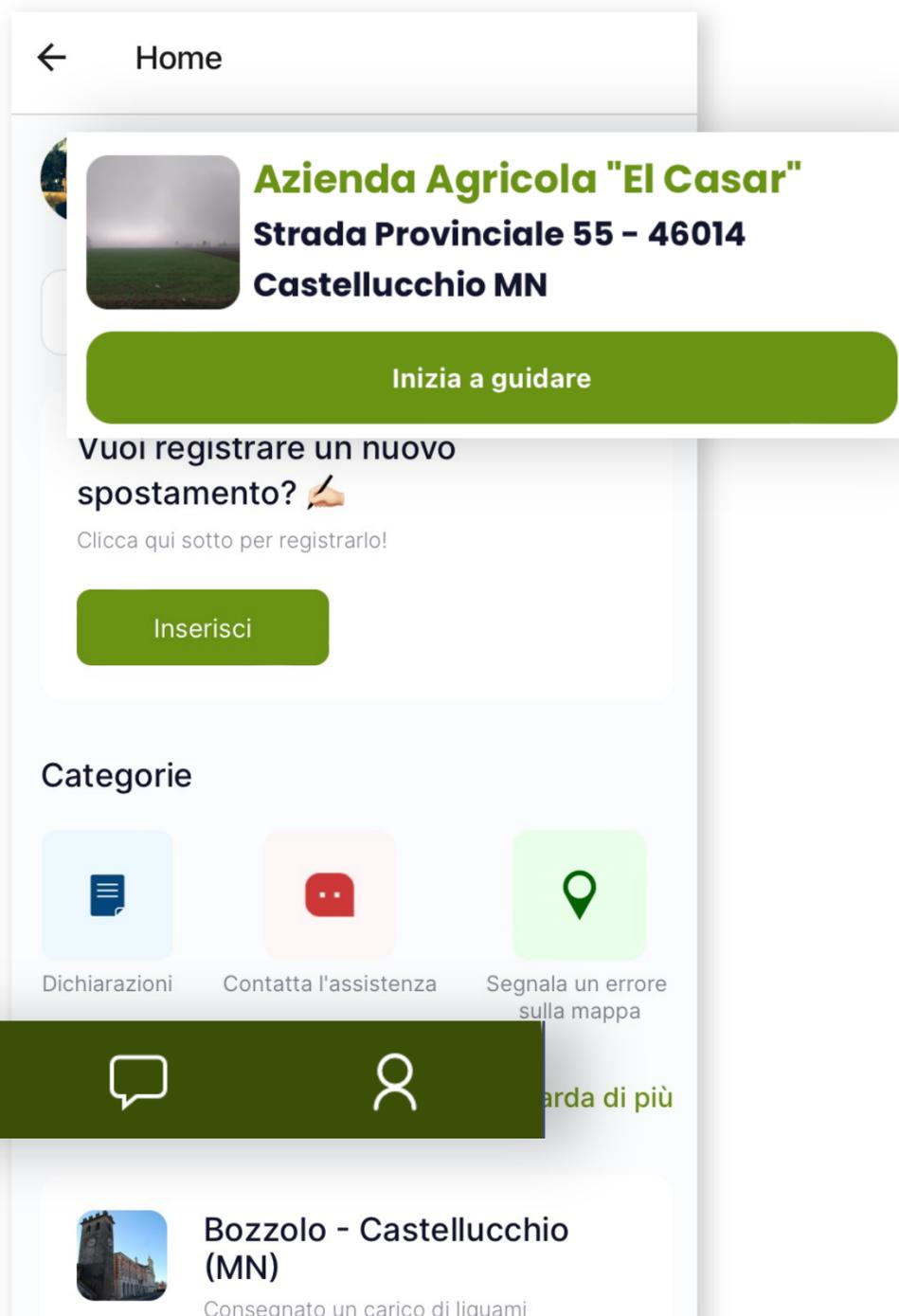
Per quanto riguarda l'applicazione serviva creare un design di facile utilizzo, abbiamo quindi diviso lo studio dell'UX design in 4 step:

Interfaccia Intuitiva: Una navigazione chiara e logica per trovare facilmente ciò di cui si ha bisogno.

Design Accattivante: Un'interfaccia moderna e piacevole alla vista per un'esperienza visiva appagante.

Facilità d'Uso: Operazioni complesse semplificate per ridurre al minimo la curva di apprendimento.

Supporto Utente: un supporto utente per segnalazione di bug o problemi.



USER EXPERIENCE



TecnoTerr
tecnologia e territorio

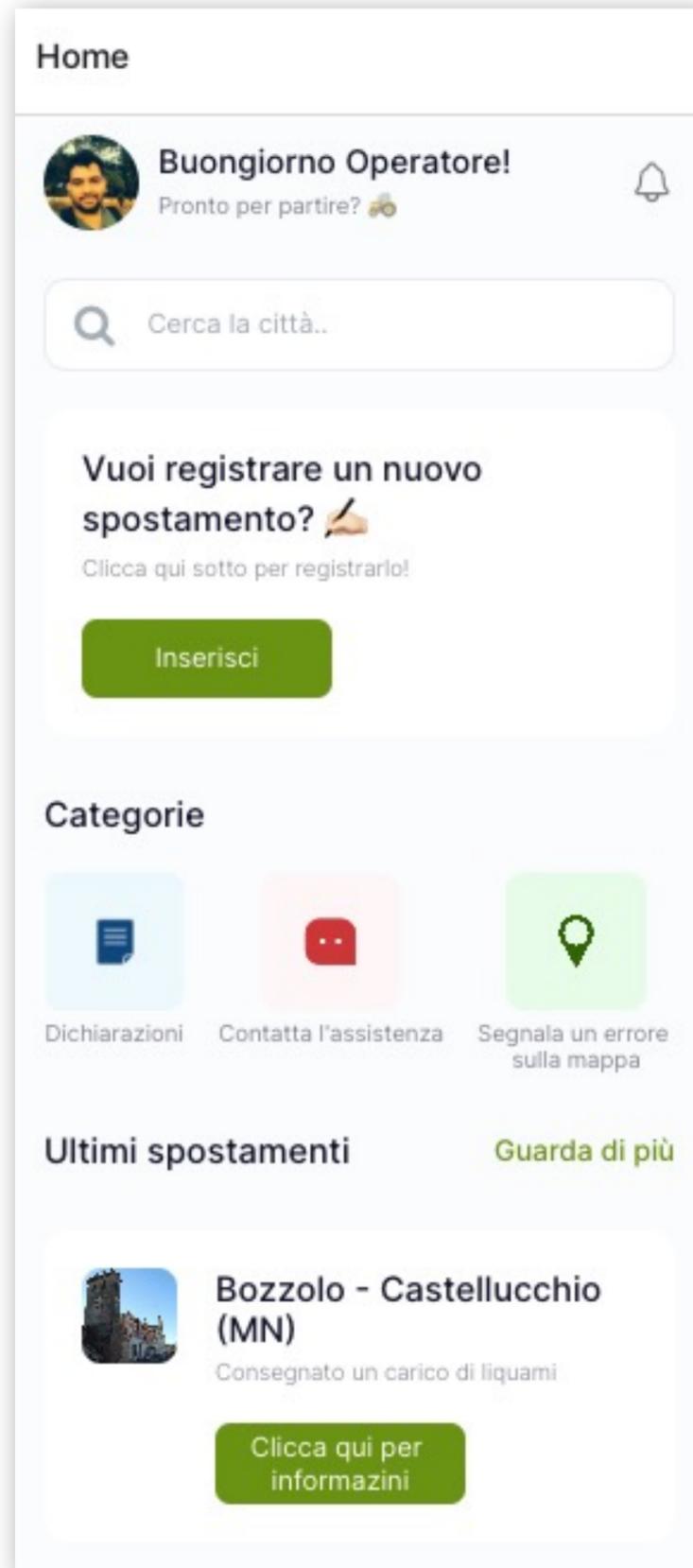
Email
@ JaneDoh@email.com

Password
MySecretPassword

Accedi

Hai bisogno d'aiuto? Contattaci!

USER EXPERIENCE



USER EXPERIENCE

←

Inserisci i tuoi dati!

Compila tutti i campi del modulo qua in basso

Nome

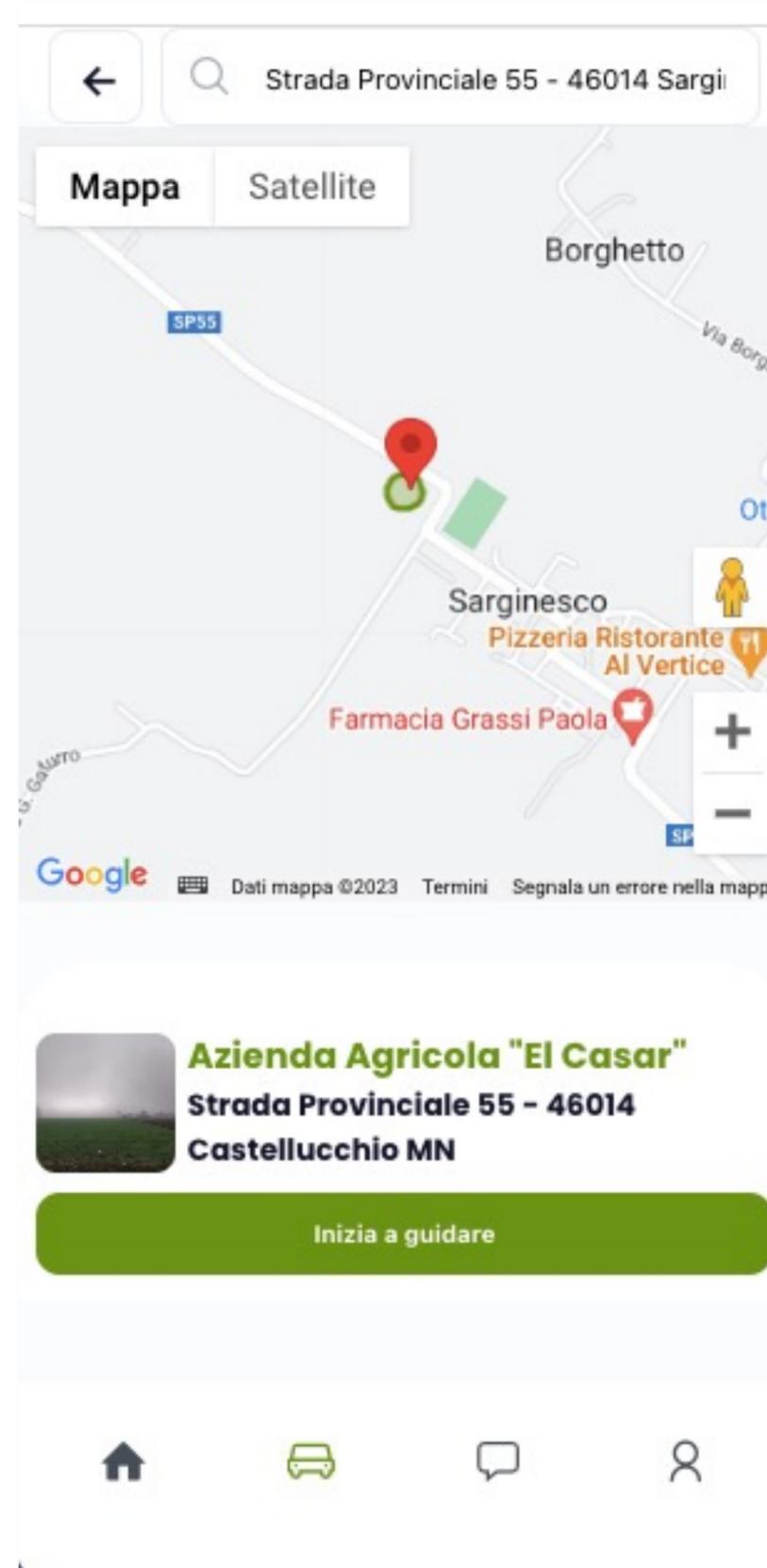
Numero di targa

Cosa trasporti?

Dove ti trovi?

Conferma

USER EXPERIENCE



USER EXPERIENCE

