

Hackathon Set 2024

Team:

- Andrao Carlo
- Alessandro Marchiò



Challenge

Gestione contravvenzioni stradali



Attualmente le contravvenzioni stradali che arrivano sulla PEC aziendale, vengono smistate manualmente da varie persone che si interfacciano con altrettanti colleghi per la gestione della pratica.

Nello specifico, la persona che presidia la PEC, individua la contravvenzione la inoltra alla persona incaricata, la quale verifica la targa sul gestionale, andando a controllare:

-Anagrafica veicolo (per auto di proprietà aziendale o vendute di recente) controllando la data vendita dell'auto, si può capire se l'auto era già stata venduta al momento della contravvenzione oppure se è attualmente in uso ad un dipendente.

-Anagrafica veicoli di cortesia per vedere se è un'auto noleggiata ad un cliente officina (controllando il periodo di noleggio) si può capire se, alla data della contravvenzione, l'auto era in uso ad un cliente

Idea di base

La nostra idea consiste nella creazione di uno script che in automatico vada raccogliere le PEC delle multe e le invii ad un pannello gestione, dal quale un operatore può accedervi in modo semplice.



FLUSSO DI GESTIONE MULTE

Un processo per gestire le multe dalla loro ricezione fino alla notifica dell'interessato.



1. FILTRAGGIO PEC



2. CREAZIONE
DATABASE



3. SCRIPT DI
LETTURA



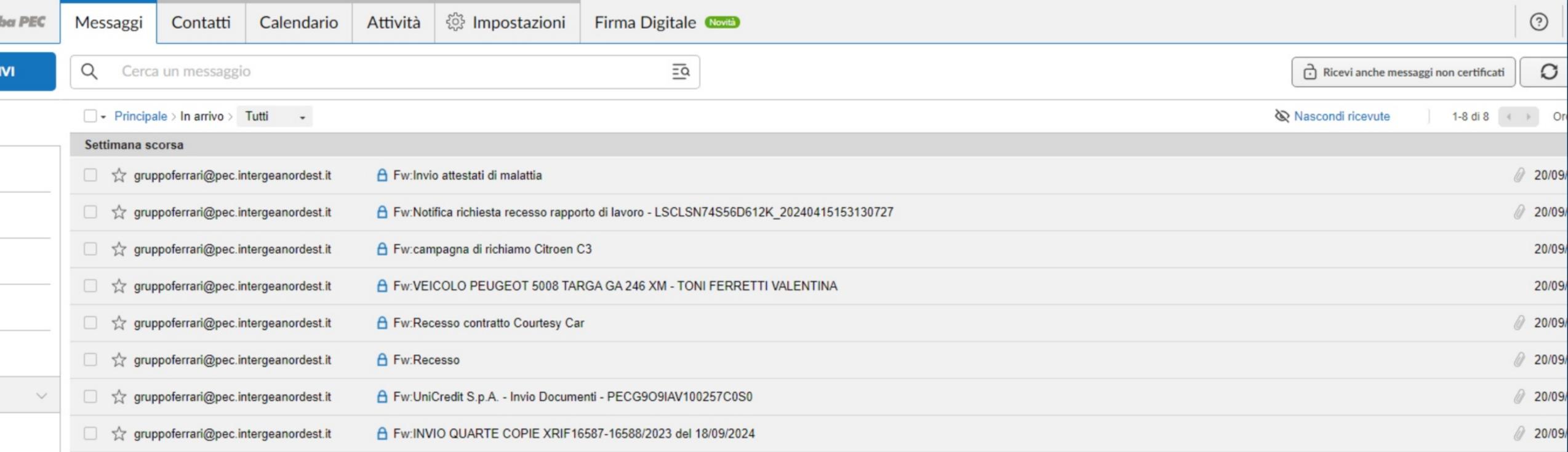
4. PANNELLO DI
GESTIONE



5. PAGAMENTO



6. NOTIFICA



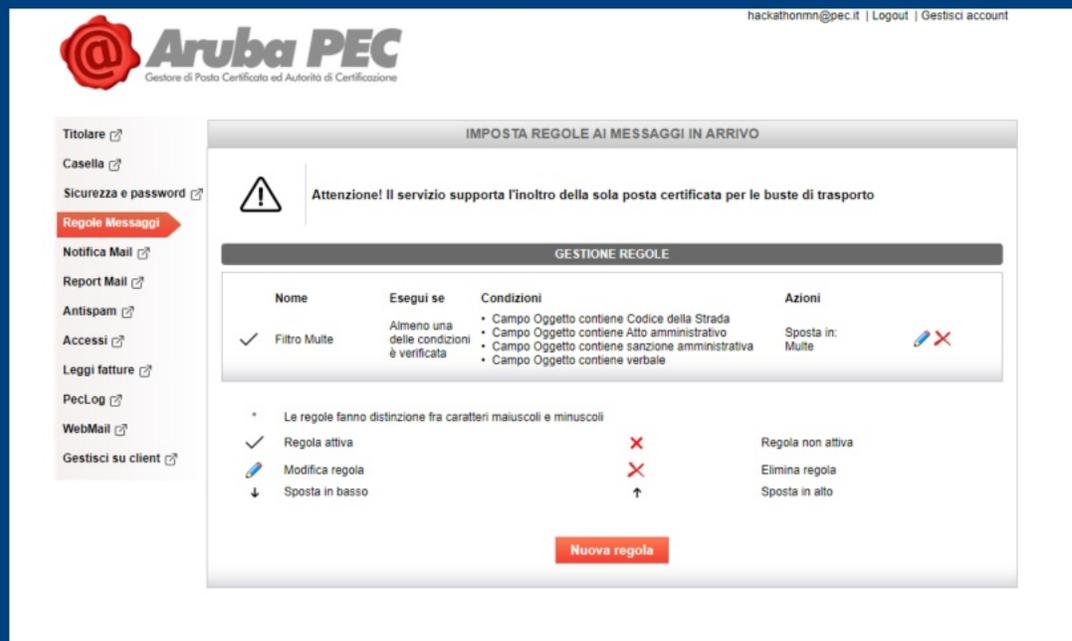
1. Filtraggio delle Email PEC

Il filtraggio delle email PEC è cruciale per raccogliere in modo efficace le molte ricevute. Attraverso una configurazione dei filtri integrati in Aruba, è possibile separare automaticamente le multe dalle PEC a noi non pertinenti.

1.a Configurazione dei filtri in Aruba

I filtri sono impostati per rilevare parole chiave che si ripetono nel campo oggetto delle multe.

Alcuni esempi sono la stringa “codice della strada” e “sanzione amministrativa”.



hackathonm@pec.it | Logout | Gestisci account

Aruba PEC
Gestore di Posta Certificata ed Autorità di Certificazione

IMPOSTA REGOLE AI MESSAGGI IN ARRIVO

⚠ Attenzione! Il servizio supporta l'inoltro della sola posta certificata per le buste di trasporto

GESTIONE REGOLE

Nome	Esegui se	Condizioni	Azioni
✓ Filtro Multe	Almeno una delle condizioni è verificata	<ul style="list-style-type: none">• Campo Oggetto contiene Codice della Strada• Campo Oggetto contiene Atto amministrativo• Campo Oggetto contiene sanzione amministrativa• Campo Oggetto contiene verbale	Sposta in: Multe  

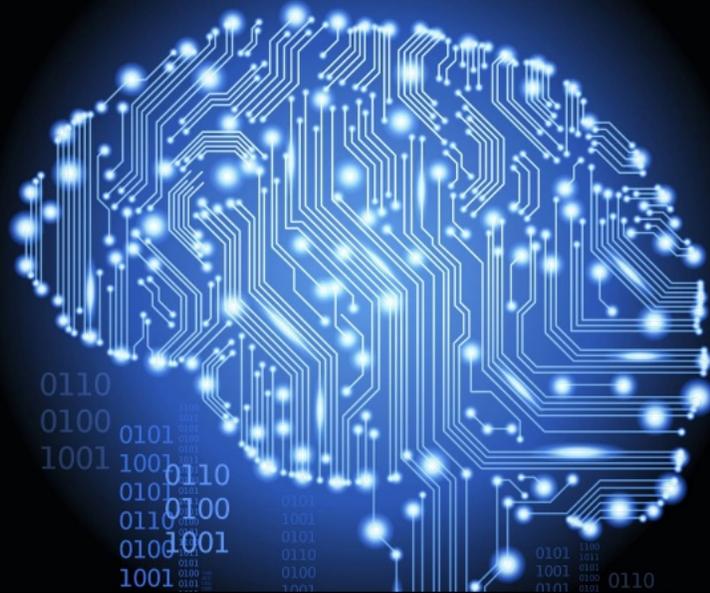
* Le regole fanno distinzione fra caratteri maiuscoli e minuscoli

- ✓ Regola attiva 
-  Modifica regola 
- ↓ Sposta in basso 

Regola non attiva 
Elimina regola 
Sposta in alto 

Nuova regola

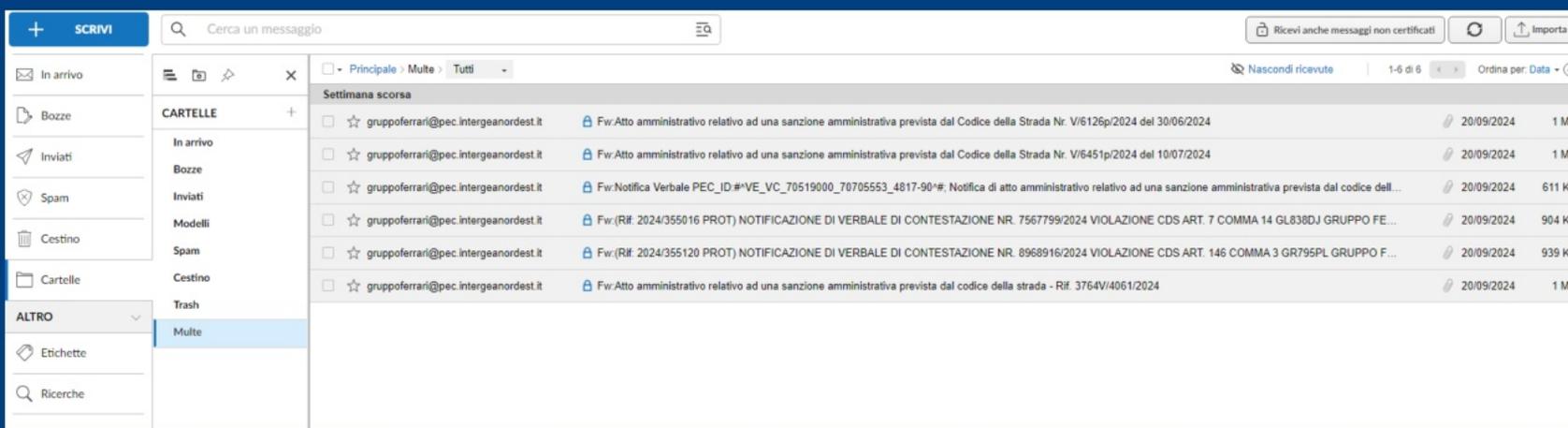
1.b Osservazioni sui filtri



I filtri sono stati realizzati manualmente analizzando attentamente le poche email PEC di esempio ricevute a disposizione. Si può prevedere l'implemento di bot intelligenti che si allenano nel riconoscimento da PEC di molte ricevute in passato.

1.c Creazione della cartella "Multe"

Le multe individuate vengono automaticamente trasferite in una cartella separata chiamata 'Multe'. Questo sistema di archiviazione non solo separa le multe dalle altre email, ma ci tornerà utile più tardi nella realizzazione dello script.

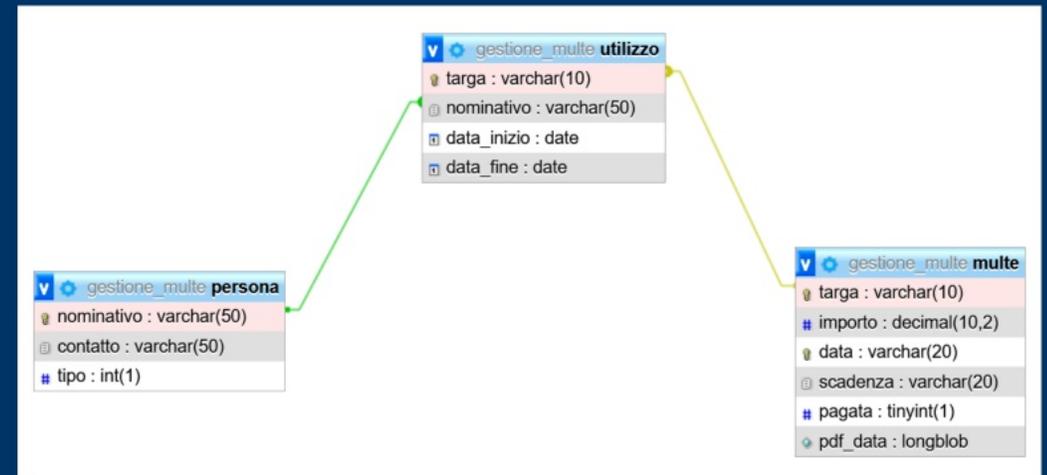


2. Creazione del Database

La creazione di un database è fondamentale per gestire in modo efficiente i dati relativi alle multe. Abbiamo utilizzato phpMyAdmin, uno strumento per la gestione di database MySQL, è possibile strutturare e organizzare le informazioni delle multe in tabelle specifiche, facilitando così l'accesso e l'analisi dei dati.

2.a Simulazione del database aziendale

Per testare le funzionalità, sono state create delle tabelle fittizie oltre a quella per le multe in modo da simulare il comportamento del database reale. Questo database simula informazioni sulle auto e sugli utenti, permettendoci il testing del nostro progetto.



2b. Tabella MULTE



A screenshot of a database table definition for 'gestione_multe multe'. The table has six columns: 'targa' (varchar(10), primary key), 'importo' (decimal(10,2), indexed), 'data' (varchar(20), primary key), 'scadenza' (varchar(20), indexed), 'pagata' (tinyint(1), indexed), and 'pdf_data' (longblob). The 'targa' column is highlighted in pink.

Column Name	Data Type	Constraints
targa	varchar(10)	Primary Key
importo	decimal(10,2)	Indexed
data	varchar(20)	Primary Key
scadenza	varchar(20)	Indexed
pagata	tinyint(1)	Indexed
pdf_data	longblob	

La tabella MULTE è progettata per contenere dati critici riguardanti le multe ricevute.

Include campi come:

- TARGA
- IMPORTO
- DATA_VIOLAZIONE
- SCADENZA
- PAGATA
- PDF

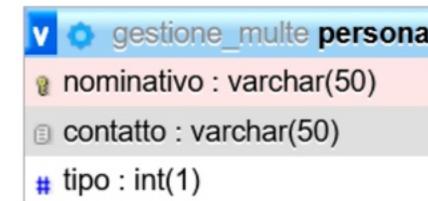
2.c Tabella UTILIZZO e PERSONA

La tabella UTILIZZO contiene informazioni su chi ha utilizzato il veicolo in un determinato periodo di tempo.

La tabella PERSONA contiene i dati delle persone che hanno utilizzato le auto con relativi dati di contatto e tipo (Aziendale, Noleggio, Auto vendita).



Column Name	Data Type
targa	varchar(10)
nominativo	varchar(50)
data_inizio	date
data_fine	date



Column Name	Data Type
nominativo	varchar(50)
contatto	varchar(50)
tipo	int(1)

3. Script Python per il Download delle multe

Per la raccolta dei dati delle multe dalla PEC abbiamo realizzato uno script Python che si connette ad Aruba per scaricare automaticamente i verbali in formato PDF. Abbiamo scelto Python perché la gestione delle stringhe ci risultava più semplice e perché presenta diverse librerie che fanno al caso nostro.

3.a Connessione ad Aruba

```
9 # CREDENTIALI PEC (inserirle qui o recuperare da un file di configurazione)
10 EMAIL = "hackathonMN@pec.it"
11 PASSWORD = "Hacka250924!"
12 SERVER = "imaps.pec.aruba.it"
13 PORT = 993
14
```

Lo script Python stabilisce una connessione a Aruba utilizzando una libreria specifica.

Questo passaggio è cruciale per accedere automaticamente alla cartella contenente i verbali delle multe.

3.b Download dei file PDF

Una funzione ci permette di scaricare tutti i verbali in formato PDF dalla cartella "Multe" della PEC.

I file non pertinenti, quindi documenti con estensioni diverse, vengono ignorati. Questo processo di download automatico semplifica notevolmente la raccolta dei dati.

I file dei verbali sono salvati in una nostra cartella in locale.

```
9 def scarica_allegati_pec(email_address, password, server, port, folder, download_folder):
10     # Crea la cartella se non esiste
11     if not os.path.exists(download_folder):
12         os.makedirs(download_folder)
13
14     # Connessione alla casella di posta IMAP
15     mail = imaplib.IMAP4_SSL(server, port)
16     mail.login(email_address, password)
17
18     # Seleziona la cartella della posta in arrivo
19     mail.select(folder)
20
21     # Cerca tutte le email (modifica il filtro come necessario)
22     status, messages = mail.search(None, 'ALL')
23     messages = messages[0].split()
24
25     #print(f"Trovate {len(messages)} email.")
```

3.c Decodifica e rinominazione dei file

I file PDF scaricati presentano un nome codificato in UTF-8.

Lo script provvede a decodificarli con una funzione e a rinominarli secondo un formato creato da noi:

"YYYYMMGG_HH-MM_ID_verbale.pdf",
ottimizzando così la gestione e il riconoscimento dei documenti.

```
27 # Funzione per decodificare il nome del file
28 def decode_filename(filename):
29     # Verifica se il nome del file è codificato
30     if filename.startswith("?"):
31         decoded_parts = email.header.decode_header(filename)
32         # Decodifica ogni parte
33         return ''.join(
34             part.decode(encoding if encoding else 'utf-8') if isinstance(part, bytes) else part for
35             part, encoding in decoded_parts)
36     return filename
37
```

```
45 with pdfplumber.open("multeAllegati/"+f) as pdf:
46     for page_num, page in enumerate(pdf.pages):
47         text = page.extract_text() # Estrai il testo dalla pagina
48         # Cerca tutte le date nella pagina
49         data = re.findall(datePattern, text)
50         # Verifica se sono state trovate almeno due date
51         if page_num < 1:
52             # Estrai anno e mese dalle prime due date trovate
53             anno1 = data[0].split("/")[2]
54             mese1 = data[0].split("/")[1]
55             anno2 = data[1].split("/")[2]
56             mese2 = data[1].split("/")[1]
57             # Gestione delle date, seleziona il verbale e la violazione
58             if anno1 >= anno2:
59                 data_verbale = data[0]
60                 data_violazione = data[1]
61                 if mese1 > mese2:
62                     data_verbale = data[0]
63                     data_violazione = data[1]
64             else:
65                 data_verbale = data[1]
66                 data_violazione = data[0]
```

3.d Estrazione dei dati dai PDF

Lo script analizza il contenuto di ciascun PDF utilizzando una libreria specializzata. Da ogni verbale, si estraggono dati chiave come TARGA, DATA_VIOLAZIONE, SCADENZA, IMPORTO e il FILE PDF.

3.1. Integrazione con il Database

Dallo script python ci colleghiamo al database per
l'invio dei dati raccolti dai verbali.



3.1.a Connessione al Database

La connessione al database viene effettuata tramite una libreria SQL dedicata. Questa connessione consente di comunicare efficacemente tra l'applicazione Python e il database phpMyAdmin, rendendo possibile l'esecuzione di query per l'inserimento dei dati.

3.2.b

Inserimento dei Dati Estratti

Una volta stabilita la connessione, i dati estratti dai file PDF vengono inseriti nel database. Ogni inserimento include le informazioni riportate nelle slide precedenti garantendo che ogni multa sia catalogata in modo accurato per future consultazioni.



4. Creazione della Piattaforma di Gestione

La piattaforma sviluppata in HTML e PHP consente agli operatori di visualizzare in modo semplice e intuitivo tutte le multe, facilitando la gestione e il monitoraggio delle stesse.

4.a Sviluppo in HTML e PHP

La piattaforma è costruita utilizzando HTML/CSS per la struttura e PHP per la gestione dei dati. La piattaforma comunica con il database per il recupero dei dati.

```
82 <!DOCTYPE html>
83 <html lang="it">
84 <head>
85     <meta charset="UTF-8">
86     <meta name="viewport" content="width=device-width, initial-scale=1.0">
87     <meta http-equiv="X-UA-Compatible" content="IE=edge">
88     <title>Area riservata</title>
89     <link rel="stylesheet" href="style.css">
90     <link rel="icon" type="image/x-icon" href="images/favicon_gruppo_move.png">
91 </head>
92 <body>
93     <div class="container">
94         
95         <h1 class="titolo">PANNELLO DI GESTIONE</h1>
96         <!-- Filtro per vedere solo le multe pagate o non pagate -->
97         <div class="filter-container">
98             <form method="GET" action="">
99                 <label for="filtra_multe">Filtra Multe:</label>
100                 <select name="filtra_multe" id="filtra_multe" onchange="this.form.submit()">
101                     <option value="non_pagato" <?php echo ($filtra_multe === 'non_pagato') ? 'selected' : '' ?>Non Pagate</option>
102                     <option value="pagata" <?php echo ($filtra_multe === 'pagata') ? 'selected' : '' ?>Pagate</option>
103                 </select>
104             </form>
105             <form method="post" style="display:inline;" onsubmit="showLoadingScreen()">
106                 <button type="submit" name="execute_python" class="btn-execute">Aggiorna &#8634;</button>
107             </form>
108         </div>
109         <div id="loadingScreen">
110             <div class="loader"></div>
111             Caricamento in corso, attendere...
112         </div>
113         <table>
114             <tr class="intestazione">
115                 <th>Targa</th>
116                 <th>Importo</th>
117                 <th>Data Violazione</th>
118                 <th>Scadenza</th>
119                 <th>Nominativo</th>
120                 <th>Contatto</th>
121                 <th>Tipologia</th>
122                 <th>PDF</th>
123                 <?php if ($filtra_multe !== 'pagata') : ?>
124                     <th>Azioni</th>
125                 <?php endif ?>
126                 <?php if ($filtra_multe === 'pagata') : ?> <?php /* parte aggiunta */ ?>
127                     <th>Invia Email</th>

```

4.b Interfaccia per l'Operatore Umano

MOVE
AUTO.SERVIZI.MOBILITÀ

PANNELLO DI GESTIONE

Filtra Multe:

Non Pagate ▾

Aggiorna ↻

Targa	Importo	Data Violazione	Scadenza	Nominativo	Contatto	Tipologia	PDF	Azioni
AA123BB	150.75 €	10/05/2024	2024-07-10	alessandro blu	ale@blu.it	Venduta	Visualizza	Segna come pagato ✓
GL838DJ	64.10 €	24/07/2024	2024-10-02	alessandro blu	ale@blu.it	Venduta	Visualizza	Segna come pagato ✓
GL838DJ	167.47 €	30/06/2024	2024-10-02	alessandro blu	ale@blu.it	Venduta	Visualizza	Segna come pagato ✓
GJ967GL	46.05 €	07/07/2024	2024-10-02	carlo bianchi	carlo@bianchi.it	Noleggio	Visualizza	Segna come pagato ✓
EE654FF	75.30 €	12/03/2024	2024-05-12	Elisa Colombo	elisa.colombo@example.it	Auto aziendale	Visualizza	Segna come pagato ✓
CC789DD	50.00 €	20/04/2024	2024-06-20	Martina Rinaldi	martina.rinaldi@example.it	Noleggio	Visualizza	Segna come pagato ✓

4.c Visualizzazione dei verbali e notifica dei pagamenti.

Gli operatori possono visualizzare i verbali direttamente dal sistema, grazie alla ricodifica dei file in formato PDF. Inoltre, cliccando sull'apposito bottone, viene inviata una mail all'interessato della multa.

The screenshot displays a web application interface with a table. The table has three columns: 'ologia', 'PDF', and an unlabeled column. The 'ologia' column contains the text 'nduta' in three rows. The 'PDF' column contains a blue button labeled 'Visualizza' in three rows. The unlabeled column contains a green button labeled 'Seg' in three rows. Below the table, there is a modal window titled 'Invia Email' with three rows, each containing a grey button labeled 'Invia Email'.

ologia	PDF	
nduta	Visualizza	Seg
nduta	Visualizza	Seg
nduta	Visualizza	Seg

Invia Email	
	Invia Email
	Invia Email
	Invia Email

5. Gestione dei Pagamenti

Il sistema permette di monitorare lo stato delle multe, facilitare il pagamento e notificare gli operatori riguardo le transazioni.

5a. Procedura di pagamento

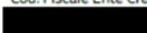
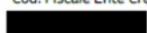
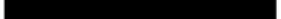
Gli operatori umani possono gestire i pagamenti delle multe visualizzando il verbale e scegliendo il metodo di pagamento che desiderano.

Dopo l'elaborazione del pagamento, è fondamentale aggiornare lo stato della multa nel sistema, segnandola come pagata per garantire trasparenza e accuratezza.

pagoPA AVVISO DI PAGAMENTO

VERBALE C.D.S. PROT. 24108123

Utilizza la porzione di avviso relativa al canale di pagamento che preferisci.

BANCHE E ALTRI	entro 5 giorni dalla notifica	BANCHE E ALTRI	dal 6° al 60° giorno dalla notifica
	Euro 122,90 Destinatario dell'avviso GRUPPO FERRARI SRL Ente Creditore COMUNE DI MODENA		Euro 173,00 Destinatario dell'avviso GRUPPO FERRARI SRL Ente Creditore COMUNE DI MODENA
Oggetto del pagamento VERBALE C.D.S. PROT. 24108123		Oggetto del pagamento VERBALE C.D.S. PROT. 24108123	
Codice CBILL	Cod. Fiscale Ente Creditore	Codice CBILL	Cod. Fiscale Ente Creditore
			
Codice Avviso		Codice Avviso	
			

Qui sopra trovi il codice QR e il codice interbancario CBILL per pagare attraverso il circuito bancario e gli altri canali di pagamento abilitati.

Automatic
payments



5b. Considerazioni sui pagamenti

Il nostro prototipo prevede un operatore umano che paga manualmente le multe. Un'implementazione futura può essere quella di automatizzare il pagamento delle multe tramite un bot automatico.

5c. Monitoraggio delle multe pagate e non pagate

Il sistema consente un monitoraggio facile delle multe, automatizzando la distinzione tra quelle pagate e non pagate. Questo report aiuta gli operatori a comprendere meglio la situazione e lo storico delle multe.



The screenshot displays the MOVE AUTO.SERVIZI.MOBILITÀ interface. At the top, the logo 'MOVE' is shown in blue, with 'AUTO.SERVIZI.MOBILITÀ' underneath. Below the logo, there is a filter section labeled 'Filtra Multe:' followed by a dropdown menu currently set to 'Pagate'. Below the filter is a table with three columns: 'Targa', 'Importo', and 'Data Violazione'. The table contains one row of data.

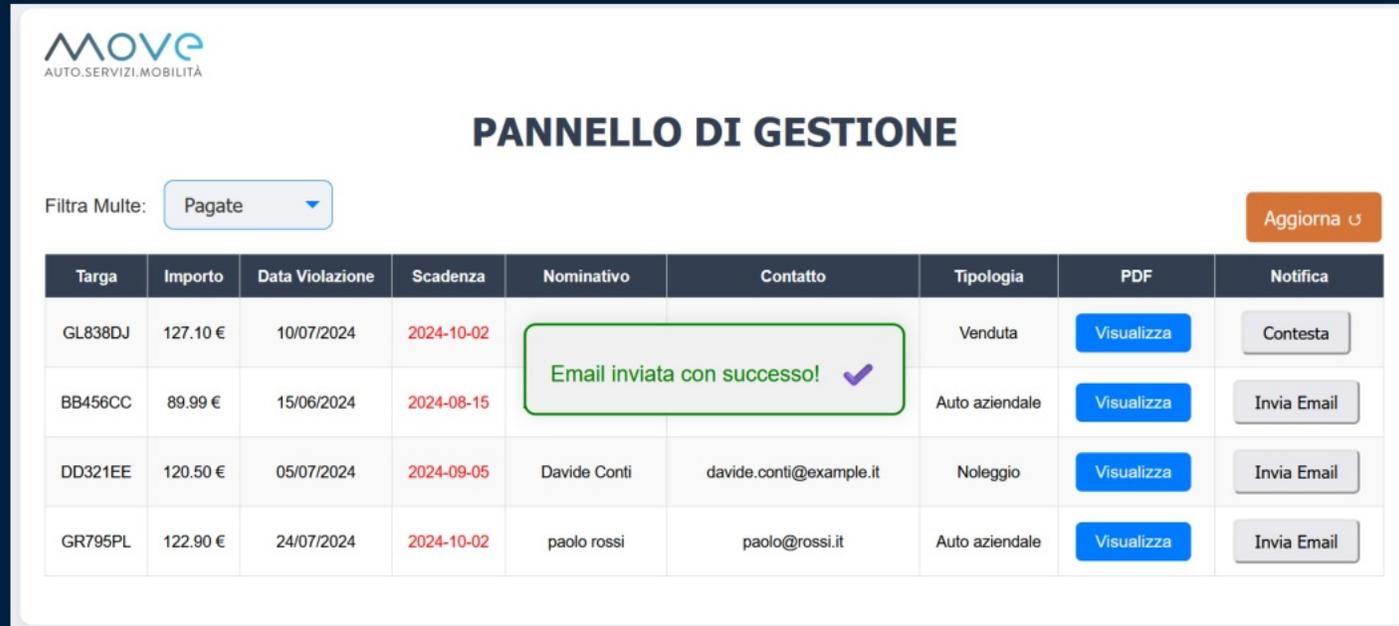
Targa	Importo	Data Violazione
GL838DJ	127.10 €	10/07/2024

6. Notifica

L'ultima parte è quella che riguarda la notifica della multa ai diretti interessati con le informazioni contenute nel verbale.

6.a Configurazione di un "server" di posta locale

Abbiamo creato una email che viene utilizzata per inviare le notifiche delle multe al click del pulsante "Invia email" nella sezione Notifica. Questo è stato possibile tramite la configurazione delle impostazioni di XAMPP.



The screenshot shows a web interface for 'MOVE AUTO.SERVIZI.MOBILITÀ'. The main heading is 'PANNELLO DI GESTIONE'. There is a filter for 'Filtra Multe:' set to 'Pagate' and an 'Aggiorna' button. A table lists four fines with columns for Targa, Importo, Data Violazione, Scadenza, Nominativo, Contatto, Tipologia, PDF, and Notifica. A green success message 'Email inviata con successo!' is overlaid on the table.

Targa	Importo	Data Violazione	Scadenza	Nominativo	Contatto	Tipologia	PDF	Notifica
GL838DJ	127.10 €	10/07/2024	2024-10-02			Venduta	Visualizza	Contesta
BB456CC	89.99 €	15/06/2024	2024-08-15			Auto aziendale	Visualizza	Invia Email
DD321EE	120.50 €	05/07/2024	2024-09-05	Davide Conti	davide.conti@example.it	Noleggio	Visualizza	Invia Email
GR795PL	122.90 €	24/07/2024	2024-10-02	paolo rossi	paolo@rossi.it	Auto aziendale	Visualizza	Invia Email

6.b Messaggio ricevuto

Notifica di atto amministrativo relativo ad una sanzione amministrativa prevista dal Codice della Strada



hackathonprova@gmail.com

a davide.conti ▾

13:10 (0 minuti fa)



Gentile Davide Conti,

La informiamo che è presente una multa a suo nome per il veicolo con targa DD321EE in data 05/07/2024.
L'importo della sanzione ammonta a €120.50.

Per ulteriori informazioni si prega di contattare l'ufficio amministrazione multe all'indirizzo amministrazione.multe@gruppomove.it.

Cordiali saluti,
Il team di gestione

← Rispondi

→ Inoltra



Hackathon Set 2024

Team:

- Andrao Carlo
- Alessandro Marchiò

