

# Esercitazione pluridisciplinare

## 1. Analisi del Tema d'Esame

**Realtà di riferimento:** Il problema riguarda lo sviluppo di una piattaforma che gestisce la condivisione di viaggi tra utenti (autisti e passeggeri). Gli utenti devono registrarsi e interagire tra loro attraverso il sito web per prenotare viaggi o offrire passaggi. Le funzionalità principali della piattaforma includono la registrazione degli utenti, la pubblicazione e la prenotazione dei viaggi, e il sistema di feedback. Inoltre, c'è un'importante gestione dei posti disponibili per ogni viaggio.

Le due categorie di utenti (autisti e passeggeri) hanno ruoli distinti, ma interagiscono in modo complementare per permettere la corretta gestione del car pooling. Le interazioni principali riguardano la prenotazione, la gestione delle disponibilità e la comunicazione tra utenti.

## 2. Casi d'Uso per la Web Application (GPO)

### 2.1 Introduzione ai Casi d'Uso

Un **caso d'uso** descrive come un attore (autista o passeggero) interagisce con il sistema per raggiungere un obiettivo. I casi d'uso sono fondamentali per progettare un'applicazione Web, poiché aiutano a identificare le funzionalità principali che il sistema deve supportare.

In questo caso, abbiamo due tipi principali di utenti: **autisti** e **passeggeri**. I casi d'uso si concentrano sulle interazioni tra questi attori e il sistema, come la registrazione, la creazione dei viaggi, la prenotazione dei viaggi, la gestione delle disponibilità dei posti, e la gestione dei feedback.

### 2.2 Casi d'Uso Principali

Di seguito, vengono descritti i **casi d'uso principali** per ciascun attore (autista e passeggero).

---

#### Casi d'Uso per l'Autista

##### 1. Registrazione dell'autista:

- **Attore:** Autista
- **Descrizione:** L'autista si registra sulla piattaforma fornendo i suoi dati personali (generalità, patente di guida, automobile, contatti). Questo processo consente all'autista di creare un profilo e di iniziare a pubblicare viaggi.
- **Flusso principale:**

1. L'autista accede alla pagina di registrazione.
2. Inserisce i dati richiesti (nome, cognome, patente, dettagli dell'automobile, numero di telefono, e-mail, fotografia).
3. Conferma l'inserimento dei dati e si registra.
4. Il sistema invia una conferma di registrazione all'autista.

## 2. Creazione di un viaggio:

- **Attore:** Autista
- **Descrizione:** L'autista crea un nuovo viaggio, specificando la città di partenza, la destinazione, la data e l'orario di partenza, il contributo economico richiesto per ogni passeggero, e le eventuali informazioni aggiuntive (ad esempio, possibilità di caricare bagagli o animali).
- **Flusso principale:**
  1. L'autista accede alla sezione "Crea viaggio".
  2. Inserisce tutte le informazioni necessarie: città di partenza, città di destinazione, data e ora, contributo economico.
  3. Seleziona il numero massimo di posti disponibili nel veicolo.
  4. Salva il viaggio nel sistema.
  5. Il viaggio è ora visibile ai passeggeri.

## 3. Gestione delle prenotazioni:

- **Attore:** Autista
- **Descrizione:** L'autista può visualizzare le prenotazioni ricevute per ciascun viaggio. Può la prenotazione di un passeggero in base alla disponibilità dei posti.
- **Flusso principale:**
  1. L'autista accede alla lista delle prenotazioni.
  2. Seleziona una prenotazione per visualizzare i dettagli del passeggero.
  3. Decide di rifiutare la prenotazione.
  4. Il sistema invia una conferma al passeggero.

#### 4. Gestione delle prenotazioni:

- **Attore:** Autista
- **Descrizione:** L'autista può visualizzare i suoi viaggi. Può chiudere le prenotazioni di un viaggio.
- **Flusso principale:**
  1. L'autista accede alla lista dei suoi viaggi.
  2. Seleziona un viaggio.
  3. Decide se chiudere le prenotazioni.

#### 5. Valutazione dei passeggeri (Feedback):

- **Attore:** Autista
- **Descrizione:** Al termine di un viaggio, l'autista può fornire un feedback sul passeggero, che può essere un voto numerico e/o un commento descrittivo.
- **Flusso principale:**
  1. L'autista accede alla sezione "Feedback".
  2. Seleziona il passeggero da valutare.
  3. Fornisce un voto numerico e scrive un commento descrittivo.
  4. Il sistema salva il feedback e lo rende visibile sia per il passeggero che per altri autisti.

#### 6. Gestione delle recensioni:

- **Attore:** Autista
- **Descrizione:** L'autista può visualizzare le recensioni dei vari utenti attraverso la loro email
- **Flusso principale:**
  1. L'autista accede alla lista delle recensioni.
  2. Visualizza le recensioni

---

## Casi d'Uso per il Passeggero

### 1. Registrazione del passeggero:

- **Attore:** Passeggero
- **Descrizione:** Il passeggero si registra sulla piattaforma fornendo i suoi dati personali (nome, cognome, documento d'identità, numero di telefono e e-mail).
- **Flusso principale:**
  1. Il passeggero accede alla pagina di registrazione.
  2. Inserisce i suoi dati personali.
  3. Il sistema invia una conferma di registrazione via e-mail.

### 2. Ricerca di viaggi:

- **Attore:** Passeggero
- **Descrizione:** Il passeggero cerca viaggi disponibili inserendo la città di partenza, la città di destinazione e la data desiderata. La piattaforma mostra tutti i viaggi che corrispondono ai criteri, esclusi quelli con prenotazioni già chiuse.
- **Flusso principale:**
  1. Il passeggero accede alla sezione "Ricerca viaggi".
  2. Inserisce la città di partenza, destinazione e la data desiderata.
  3. Il sistema mostra i viaggi disponibili con i dettagli dell'autista e la possibilità di prenotare.
  4. Il passeggero seleziona un viaggio dalla lista.
  5. Completa il processo di prenotazione.

### 3. Visualizzazione della prenotazione e notifica di accettazione:

- **Attore:** Passeggero

- **Descrizione:** Dopo aver ricevuto l'accettazione dell'autista, il passeggero riceve una conferma della prenotazione, con tutti i dettagli del viaggio (data, ora, autista, automobile, ecc.).
- **Flusso principale:**
  1. Il sistema invia una conferma via e-mail al passeggero.
  2. Il passeggero riceve tutti i dettagli della prenotazione.

#### 4. **Feedback post-viaggio:**

- **Attore:** Passeggero
- **Descrizione:** Dopo aver completato il viaggio, il passeggero può lasciare un feedback sull'autista, che consiste in un voto numerico e un commento.
- **Flusso principale:**
  1. Il passeggero accede alla sezione "Feedback".
  2. Seleziona l'autista da valutare.
  3. Fornisce un voto numerico e scrive un commento.
  4. Il sistema salva il feedback e lo rende visibile sia per l'autista che per altri passeggeri.

#### 5. **Gestione delle prenotazioni:**

- **Attore:** Passeggero
- **Descrizione:** Il passeggero può visualizzare le recensioni dei vari autisti attraverso la loro email
- **Flusso principale:**
  1. L'autista accede alla lista delle recensioni.
  2. Visualizza le recensioni

---

### 2.3 Mappa del Sito

Una **mappa del sito** è una rappresentazione grafica delle pagine principali della piattaforma e di come sono collegate tra loro. Di seguito viene proposta una mappa del sito per la piattaforma di carpooling:

1. Homepage
2. |--- Login
3. |--- Registrazione autista
4. |--- Autista (Dashboard)
5. | |--- Crea Viaggio
6. | |--- Accetta prenotazioni
7. | |--- Chiudi prenotazioni
8. | |--- Aggiungi recensioni
9. | |--- Visualizza recensioni
10. |
11. |--- Registrazione passeggero
12. | |--- Cerca viaggio
13. | |--- Visualizza prenotazione
14. | |--- Valuta viaggio
15. | |--- Visualizza recensioni

#### **Pagine principali:**

- **Homepage:** punto di ingresso alla piattaforma, con la possibilità di registrarsi o accedere.
- **Registrazione/Login:** pagina per la registrazione e il login degli utenti (autisti e passeggeri).
- **Autista (Dashboard):** pagina che consente agli autisti di gestire i loro viaggi, visualizzare le prenotazioni, e visualizzare/gestire i feedback.
- **Passeggero (Dashboard):** pagina che consente ai passeggeri di cercare viaggi, prenotare, e visualizzare/inserire i feedback.

### **3. Analisi dei Web Services per lo sviluppo dell'applicazione (TEP)**

I **Web Services** sono componenti fondamentali per la comunicazione tra i vari sistemi in un'applicazione Web. Nell'ambito di una piattaforma di carpooling, i Web Services consentono di eseguire operazioni come la registrazione degli utenti, la gestione delle prenotazioni, la ricerca dei viaggi e l'invio di feedback. Questi servizi devono essere progettati in modo da essere scalabili, sicuri e facili da integrare con il front-end della piattaforma.

#### **3.1 Funzionalità principali dei Web Services**

Analizzando la piattaforma proposta, possiamo identificare diversi **servizi principali** che devono essere sviluppati come Web Services. Ognuno di questi servizi gestisce una specifica parte dell'applicazione e interagisce con il database per recuperare, aggiornare o eliminare i dati.

### 3.2 Elenco dei Web Services Necessari

#### Autenticazione e Sessione (Comune)

- **POST /api/register.php**
    - Registrazione utente (con ruolo: passeggero o autista)
  - **POST /api/login.php**
    - Login utente (restituisce token/sessione)
  - **POST /api/logout.php**
    - Logout e distruzione sessione
  - **GET /api/user.php**
    - Recupera informazioni utente loggato
- 

#### Gestione Viaggi (Autista)

- **POST /api/viaggi/crea.php**
  - Creazione nuovo viaggio
- **GET /api/viaggi/autista.php**
  - Lista viaggi pubblicati dall'autista loggato
- **POST /api/viaggi/modifica.php?id=ID**
  - Modifica viaggio (se non ancora prenotato)
- **POST /api/viaggi/chiudi.php?id=ID**
  - Chiudi viaggio (es. viaggio completato)
- **DELETE /api/viaggi/elimina.php?id=ID**

- Elimina viaggio (se non ancora prenotato)

---

## Ricerca e Prenotazione (Passeggero)

- **GET /api/viaggi/cerca.php?partenza=...&destinazione=...&data=...**
  - Ricerca viaggi disponibili
- **GET /api/viaggi/dettagli.php?id=ID**
  - Dettagli viaggio singolo
- **POST /api/prenotazioni/invia.php**
  - Invia richiesta di prenotazione
- **GET /api/prenotazioni/mie.php**
  - Mostra tutte le prenotazioni dell'utente

---

## Gestione Prenotazioni (Autista)

- **GET /api/prenotazioni/ricevute.php**
  - Mostra prenotazioni ricevute per i viaggi dell'autista
- **POST /api/prenotazioni/accetta.php?id=ID**
  - Accetta prenotazione
- **POST /api/prenotazioni/rifiuta.php?id=ID**
  - Rifiuta prenotazione
- **POST /api/prenotazioni/rimuovi.php?id=ID**
  - Rimuove un passeggero da un viaggio accettato

---

## Recensioni



- **POST /api/recensioni/invia.php**
  - Invia recensione (passeggero → autista, dopo viaggio chiuso)
  - Invia recensione (autista → passeggero, dopo viaggio chiuso)
- **GET /api/recensioni/autista.php?id=ID\_AUTISTA**
  - Recupera recensioni per autista specifico
- **GET /api/recensioni/passeggero.php?id=ID\_PASSEGGERO**
  - Recupera recensioni per passeggero specifico

### 3.3 Tecnologie per i Web Services

Per implementare questi Web Services, si potrebbero utilizzare le seguenti tecnologie:

- **Protocollo: RESTful API** utilizzando HTTP come protocollo di comunicazione, che è semplice e ampiamente utilizzato per la realizzazione di Web Services.
- **Formato di Dati: JSON** è il formato di scambio dati preferito per le API REST, poiché è leggero, leggibile e facilmente integrabile in diverse piattaforme.
- **Autenticazione:** Per proteggere le API e assicurarsi che solo utenti autenticati possono accedere alle risorse, si potrebbe utilizzare **JWT (JSON Web Token)** per l'autenticazione basata su token.

### 3.4 Sicurezza dei Web Services

La sicurezza è fondamentale, quindi dovremmo implementare misure come:

- **Criptazione delle password:** Utilizzo di un metodo come BCrypt per criptare la password e renderla più sicura

## 4. Implementazione dell'Applicazione

Per l'implementazione del sistema, la piattaforma dovrebbe essere sviluppata utilizzando una combinazione di tecnologie client e server.

**Frontend (lato client):**

- **HTML/CSS/JavaScript:** per la creazione delle pagine web e l'interattività con gli utenti.

- **Responsive design:** per assicurarsi che la piattaforma sia fruibile su tutti i dispositivi, da desktop a smartphone, utilizzando **CSS Grid**.

#### **Backend (lato server):**

- **Database:** Un database relazionale come **MySQL** o **PostgreSQL** per memorizzare i dati degli utenti, i viaggi, le prenotazioni e i feedback.
- **Autenticazione:** Usare **Sessioni PHP** (`session_start()`) per la gestione dell'autenticazione degli utenti.
- **Integrazione email:** Servizio come **Nodemailer** per inviare le email di conferma e promemoria.

## **5. Considerazioni Finali**

L'implementazione della piattaforma di car pooling dovrebbe garantire che:

- Gli utenti possano interagire facilmente con la piattaforma.
- I dati sensibili siano gestiti in modo sicuro.
- L'interfaccia sia responsive e fruibile da ogni dispositivo.
- Il sistema di feedback sia ben integrato per garantire la qualità del servizio.