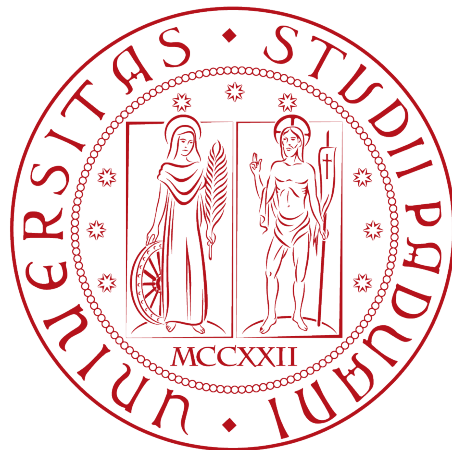


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



Implementazione Maschera Front-End per il Progetto TripHippie: Una Web App di Car Sharing

*Implementazione Maschera Front-End per il Progetto TripHippie: Una
Web App di Car Sharing*

Relatore

Prof. Claudio Palazzi

Laureando

Davide Porporati

ANNO ACCADEMICO 2022-2023

Daide Porporati: *Implementazione Maschera Front-End per il Progetto TripHippie:
Una Web App di Car Sharing*, Tesi di laurea triennale, © Dicembre 2023.

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di tirocinio, della durata di 12 settimane, dal laureando Davide Porporati presso l'azienda Synclab S.r.l. L'obiettivo del programma di stage era di impegnare lo studente nello sviluppo di un'applicazione per il car sharing. Trattandosi di un progetto di notevoli dimensioni e già avviato al momento dell'inizio del periodo di stage, l'analisi si è concentrata sui componenti già realizzati nell'ambito dello stage ed è stata fornita una prima illustrazione del prodotto finale da realizzare.

Indice

1	Introduzione	1
1.1	L'azienda	1
1.1.1	Descrizione	1
1.1.2	La mia scelta	1
1.2	Descrizione del progetto	2
1.2.1	Organizzazione del testo	2
1.2.2	Convenzioni tipografiche	2
2	Pianificazione dello stage	3
2.1	Introduzione al progetto	3
2.2	Obiettivi formativi	4
2.3	Pianificazione del lavoro	5
2.4	Aggiornamenti del piano di lavoro	7
3	Strumenti e Tecnologie	9
3.1	Linguaggi	9
3.1.1	TypeScript	9
3.1.2	HTML	9
3.1.3	CSS	10
3.2	Framework e librerie	10
3.2.1	Angular	10
3.2.2	Spring Boot	11
3.3	Strumenti di supporto	13
3.3.1	Visual Studio Code	13
3.3.2	Intellij	13
3.3.3	Swagger	13
3.3.4	Figma	13
3.3.5	Angular Material	14
3.3.6	Postegre SQL	14
4	Progettazione	17
4.1	Design Pattern Utilizzati	17
4.2	Componenti principali del sistema	19
4.3	Attori	19
4.4	Use Case	20
4.4.1	UC01 - Visualizzazione pagina "Esplora Viaggi"	20
4.4.2	UC02 - Visualizzazione pagina "Area Personale"	20
4.4.3	UC03 - Visualizzazione lista viaggi	20

4.4.4	UC03.1 - Visualizzazione singolo viaggio	21
4.4.5	UC04 - Modifica viaggio	22
4.4.6	UC05 - Elina viaggio	22
4.4.7	UC06 - Partecipa	23
4.4.8	UC07 - Disiscriviti	23
4.4.9	UC08 - Visualizzazione pagina "I miei Viaggi"	23
4.4.10	UC08.1 - Visualizzazione lista viaggi organizzati	24
4.4.11	UC08.1.1 - Visualizzazione elemento viaggio organizzato	25
4.4.12	UC08.2 - Visualizzazione lista viaggi partecipati	25
4.4.13	UC08.2.1 - Visualizzazione elemento viaggio partecipato	26
4.4.14	UC09 - Visualizzazione pagina "Inizia Viaggio"	26
4.4.15	UC10 - Visualizzazione pagina "Modifica Profilo"	27
4.4.16	UC11 - Inserimento campi testuali	27
4.4.17	UC12 - Crea nuovo viaggio	28
4.5	Requisiti	28
4.5.1	Funzionali	28
4.6	Sviluppo Applicazione Web	30
4.6.1	Home	30
4.6.2	Login	30
4.6.3	Explorepage	30
4.6.4	Details-trip	31
4.6.5	My-trip	32
4.6.6	Start-trip	32
4.6.7	Edit-profile	33
5	Conclusioni	35
5.1	Raggiungimento degli obiettivi	35
5.2	Punti di forza e di debolezza del progetto	36
5.3	Considerazioni finali	36
	Acronimi e abbreviazioni	37
	Glossario	39
	Sitografia	41

Elenco delle figure

1.1	Logo di <i>Synclab</i>	1
3.1	Logo di TypeScript	9
3.2	Logo HTML	10
3.3	Logo CSS	10
3.4	Logo di Angular	11
3.5	Logo di Springboot	12
3.6	Logo di Visual Studio Code	13
3.7	Logo di IntelliJ	13
3.8	Logo di Swagger	13
3.9	Logo di Figma	14
3.10	Logo di Angular Material	14
3.11	Logo di Postgre SQL	14
3.12	Database relazione di TripHippie	15
3.13	Esempio dell'utilizzo di Postgre SQL Shell	15
4.1	Struttura dell'applicazione	18
4.2	Attori	19
4.3	Navigazione su TripHippie	21
4.4	UC03	21
4.5	UC03.1	22
4.6	Dettaglio viaggio	24
4.7	UC08	24
4.8	UC08.1	25
4.9	UC08.2	26
4.10	Navigazione Area Personale	27
4.11	Pagina Inizia Viaggio	28
4.12	Home	30
4.13	Login	30
4.14	Pagina Esplora Viaggi	31
4.15	Dettaglio viaggio	31
4.16	Pagina Viaggi Organizzati	32
4.17	Pagina Viaggi Partecipati	32
4.18	Pagina Inizia Viaggio	33
4.19	Pagina Modifica Profilo	33

Elenco delle tabelle

2.1	Tabella degli obiettivi	4
2.2	Tabella delle ore previste	6
2.3	Tabella delle ore previste aggiornata	7
4.1	Tabella del tracciamento dei requisiti	29
5.1	Tabella degli obiettivi raggiunti	35

Capitolo 1

Introduzione

1.1 L'azienda

1.1.1 Descrizione

SyncLab è un'azienda innovativa che ha fatto della trasformazione digitale il proprio cavallo di battaglia. Attraverso la realizzazione di prodotti e soluzioni su misura, l'azienda opera in vari settori, tra cui Sanità, Industria, Energia, Finanza, Trasporti e Logistica. Nelle quattro sedi dislocate in Italia, l'azienda offre consulenza specialistica da oltre 20 anni nel campo dell'IT sfruttando una solida competenza in una varietà di domini tecnologici. Questo approccio consente ai clienti di affrontare con successo le sfide emergenti, come il *Big Data*, il *Cloud Computing*, l'*IoT*, la *Mobile* e la *Cyber Security*, e di capitalizzare sui nuovi trend digitali. In questo contesto, *SyncLab* si afferma come uno dei principali *system integrator* nell'ambito delle aziende ICT in Italia. Nata come [Software House](#), *SyncLab* evolve rapidamente grazie al costante sviluppo delle competenze tecnologiche, metodologiche ed applicative nel campo del software. L'azienda si distingue per la creazione di prodotti, frutto del costante impegno nella ricerca e nello sviluppo. Grazie a questi prodotti, *Sync Lab* ha gradualmente acquisito una posizione di rilievo in settori come la mobilità, la videosorveglianza e la sicurezza delle infrastrutture informatiche aziendali.



Figura 1.1: Logo di *SyncLab*

1.1.2 La mia scelta

Sono entrato in stretto contatto con l'azienda durante l'evento "StageIT 2023", durante il quale ho avuto modo di conoscere l'Ing. Fabio Pallaro. È lui che mi ha illustrato le diverse opportunità di stage offerte dall'azienda, all'interno della quale i potenziali stagisti possono selezionare progetti che rispecchino i propri interessi. La presentazione è stata persuasiva e ha stimolato fin da subito la mia curiosità. Successivamente,

attraverso un colloquio individuale effettuato nella sede di Padova nel mese di aprile, ho avuto l'opportunità di scegliere il progetto che meglio si adattava ai miei interessi, il progetto "*TripHippie*".

1.2 Descrizione del progetto

1.2.1 Organizzazione del testo

Il primo capitolo ha descritto brevemente l'azienda e la scelta del progetto.

Il secondo capitolo entra maggiormente nei dettagli del progetto di stage descrivendo la pianificazione temporale e gli obiettivi richiesti.

Il terzo capitolo descrive le tecnologie utilizzate che sono alla base del progetto.

Il quarto capitolo approfondisce l'analisi dei requisiti con i rispettivi casi d'uso analizzati.

Nel quinto e ultimo capitolo viene riportato un consuntivo delle attività svolte e una valutazione dello stage con alcune riflessioni personali.

1.2.2 Convenzioni tipografiche

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- i termini in lingua straniera o facenti parte del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Pianificazione dello stage

In questo capitolo viene presentato nel dettaglio il progetto TripHippie insieme agli obiettivi attesi e alla pianificazione delle attività da svolgere.

2.1 Introduzione al progetto

Lo scopo dello stage è quello di realizzare un'App di car sharing [Car Sharing](#). La congestione del traffico verso le località turistiche durante i weekend è il problema da affrontare. L'idea è quella di provare a pensare ad un prodotto che unisca la possibilità di car sharing all'incontro tra persone, ispirandosi all'Hippie trail degli anni '50. Oltre alle classiche funzionalità di un *car sharing* l'app fornirà:

- profilazione utenti;
- disponibilità ad approfondire conoscenza tra passeggeri;
- possibilità di pranzo/cena tra le persone del gruppo in viaggio (pizza, ristorante, sushi);
- possibilità di mandare inviti di viaggio privati tra utenti;
- scelta (consapevole) di fare uso di sostanze alcoliche o stupefacenti durante il percorso.

Il percorso di tirocinio prevede anche la parte di implementazione dove sono state sviluppate le maschere di *front-end* per la login/registrazione dell'utente, l'inserimento di proposte di viaggio e la loro visualizzazione.

2.2 Obiettivi formativi

Si farà riferimento ai requisiti utilizzando le seguenti notazioni:

- **OB** per i requisiti obbligatori, vincolanti in quanto obiettivo primario, richiesto dal committente;
- **DE** per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- **OP** per i requisiti opzionali, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da un numero, identificativo del requisito.

Nella tabella 2.1 Si elencano lo svolgimento dei seguenti obiettivi:

Obbligatorio	
OB1	Acquisizione competenze sulle tematiche sopra descritte
OB2	Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma
OB3	Portare a termine le implementazioni previste con una percentuale di superamento pari al 80% (equivalente alla maschera d'inserimento Viaggio e relativo service con le chiamate al <i>back-end</i>)
Desiderabile	
DE1	Portare a termine le implementazioni previste con una percentuale di superamento pari al 100% (equivalente alla maschera di Inserimento, gestione e visualizzazione Viaggi e relativi <i>service</i> con le chiamate al <i>back-end</i>)
Opzionale	
OP1	Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce

Tabella 2.1: Tabella degli obiettivi

2.3 Pianificazione del lavoro

La pianificazione, in termini di quantità di ore lavorative, è stata così distribuita:

N° settimana	Descrizione attività
Settimana 1 (20 ore)	<ul style="list-style-type: none">• Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;• Presentazione strumenti di lavoro per la condivisione del materiale di studio e per la gestione dell'avanzamento;• Condivisione scaletta di argomenti;
Settimana 2 (20 ore)	<ul style="list-style-type: none">• Ripasso concetti Metodologia Agile/Scrum;• Ripasso del linguaggio Java SE;• Ripasso concetti Web (Servlet, servizi Rest, Json ecc.);• Studio principi generali di Spring Core (IOC, Dependency Injection).
Settimana 3 (20 ore)	<ul style="list-style-type: none">• Studio SpringBoot.
Settimana 4 (20 ore)	<ul style="list-style-type: none">• Studio Spring Data/DataRest.
Settimana 5 (20 ore)	<ul style="list-style-type: none">• Ripasso linguaggio Javascript.
Settimana 6 (20 ore)	<ul style="list-style-type: none">• Studio del linguaggio TypeScript.
Settimana 7 (20 ore)	<ul style="list-style-type: none">• Studio piattaforma NodeJS e AngularCLI.

Settimana 8 (20 ore)	<ul style="list-style-type: none"> • Studio framework Angular.
Settimana 9 (20 ore)	<ul style="list-style-type: none"> • Analisi e studio del progetto TripHippie.
Settimana 10 (20 ore)	<ul style="list-style-type: none"> • Progettazione ed implementazione della nuova maschera di accesso.
Settimana 11 (20 ore)	<ul style="list-style-type: none"> • Progettazione ed implementazione nuova maschera "Inserimento e gestione Viaggio".
Settimana 12 (20 ore)	<ul style="list-style-type: none"> • Scrittura dei service (su front-end) di chiamata al <i>back-end</i>.
Settimana 13 (20 ore)	<ul style="list-style-type: none"> • Progettazione ed implementazione nuova maschera "Visualizzazione Viaggi".
Settimana 14 (20 ore)	<ul style="list-style-type: none"> • Scrittura dei service (su front-end) di chiamata al <i>back-end</i>.
Settimana 15 (20 ore)	<ul style="list-style-type: none"> • Termine integrazioni e collaudo finale.
Settimana 16 (20 ore)	<ul style="list-style-type: none"> • Termine integrazioni e collaudo finale.

Tabella 2.2: Tabella delle ore previste

2.4 Aggiornamenti del piano di lavoro

A conclusione della decima settimana, il mio impegno di stage è passato da un regime *part-time* a un impegno *full-time*, con una durata di 40 ore settimanali. Di conseguenza, il periodo di stage inizialmente pianificato per 16 settimane è stato ridotto a 13 settimane, di cui 10 in modalità *part-time* e 3 a tempo pieno. Di seguito il piano di lavoro delle rimanenti 3 settimane *full-time*:

Settimana 11 (40 ore)	<ul style="list-style-type: none"> • Progettazione ed implementazione nuova maschera "Inserimento e gestione Viaggio". • Scrittura dei service (su <i>front-end</i>) di chiamata al <i>back-end</i>.
Settimana 12 (40 ore)	<ul style="list-style-type: none"> • Progettazione ed implementazione nuova maschera "Visualizzazione Viaggi". • Scrittura dei service (su <i>front-end</i>) di chiamata al <i>back-end</i>.
Settimana 13 (40 ore)	<ul style="list-style-type: none"> • Termine integrazioni e collaudo finale.

Tabella 2.3: Tabella delle ore previste aggiornata

Capitolo 3

Strumenti e Tecnologie

In questo capitolo viene data una descrizione dettagliata di tutti i linguaggi, framework, librerie e altri strumenti utilizzati durante il tirocinio

3.1 Linguaggi

3.1.1 TypeScript

Il linguaggio [TypeScript](#) è un linguaggio di programmazione orientato agli oggetti realizzato da Microsoft. Si tratta di un super-set di Javascript ed è utilizzato dal *framework Angular* come linguaggio principale per lo sviluppo di siti e applicazioni *web*.



Figura 3.1: Logo di TypeScript

3.1.2 HTML

[Hyper Text Markup Language \(HTML\)](#) è un linguaggio di markup utilizzato per la creazione di pagine *web*. Non è un linguaggio di programmazione, ma un linguaggio di marcatura che definisce la struttura e il contenuto delle pagine web. HTML fornisce elementi e tag che consentono di definire testo, immagini, link ipertestuali e altri elementi all'interno di una pagina web. È un linguaggio fondamentale per lo sviluppo web e costituisce la struttura fondamentale su cui si basano le pagine web.



Figura 3.2: Logo HTML

3.1.3 CSS

[Cascading Style Sheets \(CSS\)](#) è un linguaggio utilizzato per definire l'aspetto e la formattazione di documenti HTML. Mentre HTML definisce la struttura e il contenuto della pagina, CSS ne controlla l'aspetto visivo. Con CSS, è possibile specificare il colore, la dimensione, il *layout* e altre proprietà degli elementi HTML. Questo separa la presentazione (CSS) dal contenuto (HTML) delle pagine web, consentendo un maggiore controllo sul design e la formattazione. CSS è un componente essenziale per la progettazione di siti web e la creazione di layout "responsive" e attraenti.



Figura 3.3: Logo CSS

3.2 Framework e librerie

In questa sezione vengono presentati librerie e [Framework](#) utilizzati durante lo sviluppo dei vari componenti del prodotto

3.2.1 Angular

Angular è un *framework* scritto in TypeScript per lo sviluppo di siti o applicazioni web. L'unità fondamentale utilizzata da Angular è il [Component](#) che è una classe Typescript a cui è associato un *decorator* contenente il foglio di stile CSS e il file HTML contenente la struttura del Component stesso. Ogni Component rappresenta un elemento riutilizzabile ed indipendente dell'interfaccia del sito o dell'applicazione web. Il file HTML che contiene la struttura contiene degli elementi non standard che

sono utilizzati da Angular per rendere l'interfaccia *responsive* rispetto ai cambiamenti del modello (rappresentato dall'istanza di riferimento della classe Typescript). Tali elementi sono le *Angular directives* e contengono riferimenti a specifici parametri e/o metodi del modello. Di seguito alcuni esempi:

- `*ngFor`: ripete un certo tag per ogni elemento di un oggetto iterabile
- `*ngIf`: genera un certo tag se una determinata condizione è vera
- `{{...}}`: la direttiva di interpolazione inietta direttamente nel codice HTML il valore ottenuto da una data operazione
- `(click)`: dichiara un listener per un determinato evento che coinvolge un dato elemento della pagina
- `[(ngModel)]`: lega il campo value di un tag al valore di una variabile del modello. Ciò consente di avere costantemente consistenza tra il valore mostrato e quello realmente memorizzato.



Figura 3.4: Logo di Angular

3.2.2 Spring Boot

Spring Boot è un *framework open source* per lo sviluppo di applicazioni Java. È progettato per semplificare notevolmente il processo di creazione di applicazioni Java, fornendo un approccio "convenzione su configurazione", il che significa che molte decisioni di configurazione predefinite sono già incluse, riducendo la necessità di configurazioni personalizzate. L'unità fondamentale di sviluppo in Spring Boot è l'applicazione Spring Boot stessa, che è tipicamente costituita da una classe principale con un metodo 'main'. Spring Boot gestisce molte delle configurazioni di base necessarie per avviare l'applicazione, inclusa la configurazione del *server web embedded*. Spring Boot offre anche una serie di librerie e moduli che semplificano lo sviluppo di vari aspetti delle applicazioni, tra cui l'accesso ai *database*, la sicurezza, la messaggistica e altro ancora. Inoltre, Spring Boot promuove l'uso di annotazioni per semplificare l'iniezione delle dipendenze e la configurazione dell'applicazione. Ecco alcune caratteristiche e concetti chiave di Spring Boot:

- ****Spring Boot Starter****: Questi sono moduli preconfigurati che semplificano l'integrazione di diverse tecnologie e *framework*. Ad esempio, 'spring-boot-starter-web' semplifica la creazione di applicazioni web;

- **Auto-Configuration**: Spring Boot include automaticamente le configurazioni per molte tecnologie comuni. Ad esempio, se c'è bisogno di un database, Spring Boot può configurare automaticamente il supporto per un database in base alle dipendenze necessarie;
- **Spring Boot CLI**: È uno strumento da riga di comando che permette di creare, sviluppare e testare rapidamente applicazioni Spring Boot;
- **Embedded Servers**: Spring Boot supporta server web incorporati come Tomcat, Jetty e Undertow, eliminando la necessità di configurare manualmente un server web esterno;
- **Spring Boot Actuator**: Fornisce funzionalità di monitoraggio e gestione per le applicazioni Spring Boot, tra cui endpoint per ottenere informazioni sullo stato dell'applicazione;
- **Spring Boot DevTools**: Sono strumenti che aiutano a sviluppare applicazioni Spring Boot più velocemente, con funzionalità come il riavvio automatico dell'applicazione durante lo sviluppo.

Spring Boot è ampiamente utilizzato per sviluppare applicazioni Java aziendali, inclusi servizi web, applicazioni *RESTful*, applicazioni di microservizi e molto altro. È noto per la sua facilità d'uso e per la sua capacità di accelerare il processo di sviluppo.



Figura 3.5: Logo di Springboot

3.3 Strumenti di supporto

3.3.1 Visual Studio Code

Visual Studio Code (VSC) è un popolare ambiente di sviluppo integrato (IDE) *open-source* sviluppato da Microsoft. È noto per la sua leggerezza, estensibilità e supporto per numerosi linguaggi di programmazione, rendendolo una scelta comune tra gli sviluppatori. È stato utilizzato per elaborare codice durante la parte di *front-end*.



Figura 3.6: Logo di Visual Studio Code

3.3.2 IntelliJ

IntelliJ è un ambiente di sviluppo integrato **Integrated Development Environment (IDE)** sviluppato da JetBrains, noto per il suo eccellente supporto per la programmazione Java. Offre funzionalità avanzate di sviluppo, *debugging* e *refactoring* per una vasta gamma di linguaggi di programmazione. È stato impiegato per avviare le chiamate al *back-end*, già presenti all'inizio del periodo di stage.



Figura 3.7: Logo di IntelliJ

3.3.3 Swagger

Swagger è una *suite* di strumenti *open-source* per la progettazione, la creazione e la documentazione di **Representational State Transfer (REST) API (API REST)**. Fornisce un modo standardizzato per descrivere le API, facilitando la collaborazione tra sviluppatori e clienti. È stato prezioso nell'identificare la struttura delle funzioni nel *back-end* e nel verificare l'efficacia delle stesse.



Figura 3.8: Logo di Swagger

3.3.4 Figma

Figma è un'applicazione di progettazione e prototipazione basata su *cloud*. È ampiamente utilizzata per la progettazione di interfacce utente, consentendo la collaborazione

in tempo reale tra *designer* e sviluppatori. È stato di grande aiuto nel generare una bozza iniziale del progetto *TripHippie* e ha permesso una comprensione più approfondita delle modifiche necessarie sia nel *back-end* che nel *front-end*.



Figura 3.9: Logo di Figma

3.3.5 Angular Material

Angular Material è una libreria di componenti e stili CSS predefiniti sviluppata da *Google* per Angular. Aiuta gli sviluppatori a creare interfacce utente coerenti e attraenti per le loro applicazioni web. È stato utile perché ha fornito un insieme di componenti UI predefiniti e stili coerenti per lo sviluppo dell'applicazione web. Questo ha semplificato la creazione delle interfacce, garantendo al contempo una coerenza nel *design* e un'esperienza uniforme. Inoltre, Angular Material offre funzionalità come la gestione della responsività e l'accessibilità, rendendo più agevole lo sviluppo di applicazioni web di alta qualità.



Figura 3.10: Logo di Angular Material

3.3.6 Postgre SQL

PostgreSQL è un potente sistema di gestione di database relazionali *open-source*. È noto per la sua affidabilità, prestazioni elevate e supporto per estensioni, rendendolo una scelta popolare tra gli sviluppatori di database.



Figura 3.11: Logo di Postgre SQL

Un database relazionale è un tipo di database che organizza i dati in tabelle composte da righe e colonne, seguendo il modello di dati relazionale. In questo modello, i dati sono archiviati in tabelle, ciascuna delle quali rappresenta una singola entità o oggetto, e le relazioni tra le entità sono gestite attraverso chiavi primarie e chiavi esterne. Le tabelle sono interconnesse in modo che sia possibile recuperare, aggiornare, o inserire dati in base alle relazioni tra le diverse tabelle. Le caratteristiche principali di un

database relazionale includono la struttura tabellare, l'integrità dei dati attraverso vincoli, la possibilità di interrogare i dati utilizzando il linguaggio [Structured Query Language \(SQL\)](#), e la capacità di gestire relazioni complesse tra gli oggetti archiviati nei database. Questo modello è ampiamente utilizzato in applicazioni aziendali e software in cui è necessario gestire dati strutturati in modo organizzato e affidabile. Nella figura 3.12 vengono riportata la struttura tabellare del progetto TripHippie:

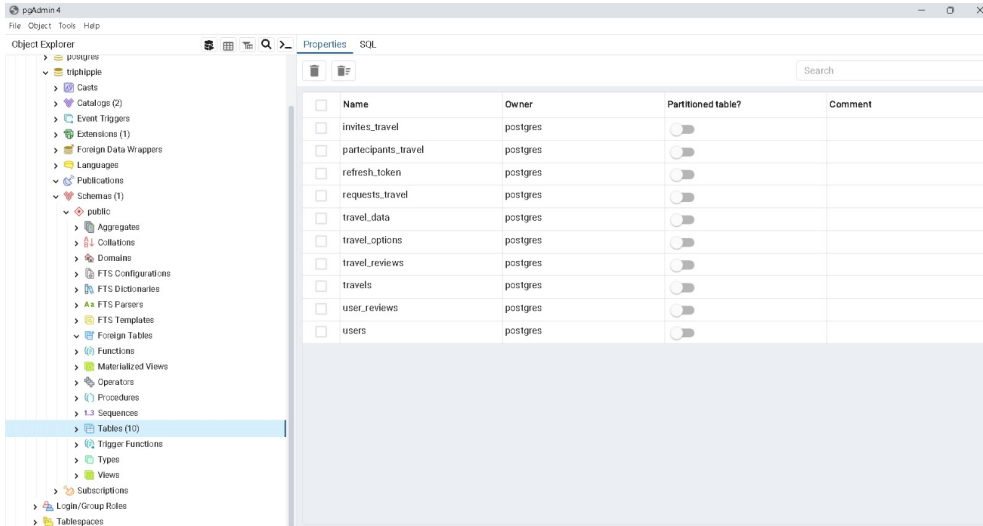


Figura 3.12: Database relazione di TripHippie

È stato molto utile per la gestione del database, grazie anche al suo SQL Shell. Di seguito un esempio per visualizzare determinati parametri degli utenti:

```

SQL Shell (psql)
Server [localhost]:
Database [postgres]: triphippie
Port [5433]:
Username [postgres]:
Inserisci la password per l'utente postgres:
psql (16.0)
ATTENZIONE: Il code page della console (850) differisce dal code page
di Windows (1252). I caratteri a 8-bit potrebbero non
funzionare correttamente. Vedi le pagine di riferimento
psql "Note per utenti Windows" per i dettagli.
Digita "help" per avere un aiuto.

triphippie=# select users.username, users.user_id, users.name, users.surname from users;
 username | user_id | name  | surname
-----+-----+-----+-----
 lrossi90 |      2 | Luca  | Rossi
 sferrari83 |      3 | Sofia | Ferrari
 Porpo46  |      1 | Davide | Porporati
(3 righe)

triphippie=#

```

Figura 3.13: Esempio dell'utilizzo di Postgre SQL Shell

Capitolo 4

Progettazione

In questo capitolo viene data una spiegazione dettagliata dei casi d'uso e dello sviluppo vero e proprio dell'applicazione web

4.1 Design Pattern Utilizzati

Il classico design pattern [Model-View-Controller \(MVC\)](#) è una struttura architeturale composta da tre componenti principali:

- **Model:** include interfacce e classi che gestiscono principalmente i dati e la logica di business dell'applicazione. È responsabile per l'accesso ai dati e la loro manipolazione, ma non contiene logica direttamente correlata all'interfaccia utente;
- **View:** rappresenta l'interfaccia utente dell'applicazione. Si occupa di visualizzare i dati al utente e raccoglie le interazioni dell'utente, ma non gestisce la logica di business o i dati stessi;
- **Controller:** funge da intermediario tra il Model e la View. Riceve gli input dall'utente attraverso la View, elabora tali input e interagisce con il Model per ottenere o aggiornare i dati. La sua principale responsabilità è coordinare l'interazione tra Model e View.

Angular, d'altra parte, adotta un design pattern leggermente diverso, che si avvicina al [Model-View-ViewModel \(MVVM\)](#). Questo modello è caratterizzato da un sistema di data binding bidirezionale tra "View" e "ViewModel". Nel contesto di Angular, le componenti corrispondono approssimativamente ai seguenti ruoli:

- **Model:** rappresenta principalmente i dati dell'applicazione, ma spesso è incorporato nel ViewModel.
- **View:** la View in Angular rappresenta l'interfaccia utente, ma in realtà è fortemente legata al ViewModel. La View riflette lo stato del ViewModel e può anche reagire dinamicamente ai cambiamenti del ViewModel.
- **ViewModel:** il ViewModel in Angular è il componente principale che gestisce la logica dell'interfaccia utente e l'interazione con il Model. È responsabile della

gestione dei dati e dell'interazione con la View. Angular sincronizza automaticamente i dati tra la View e il ViewModel, consentendo così un'interazione fluida tra di essi.

In sintesi, mentre il classico MVC separa chiaramente il Model, la View e il Controller, Angular utilizza un modello che si basa su un'architettura MVVM, in cui il ViewModel svolge un ruolo chiave nell'aggiornamento della View e nella gestione della logica di interfaccia utente. Questo approccio semplificato e leggermente diverso consente un data binding bidirezionale che semplifica lo sviluppo di applicazioni web complesse.

In figura 4.1 l'elenco dei vari Component e la struttura dell'applicativo sviluppato:

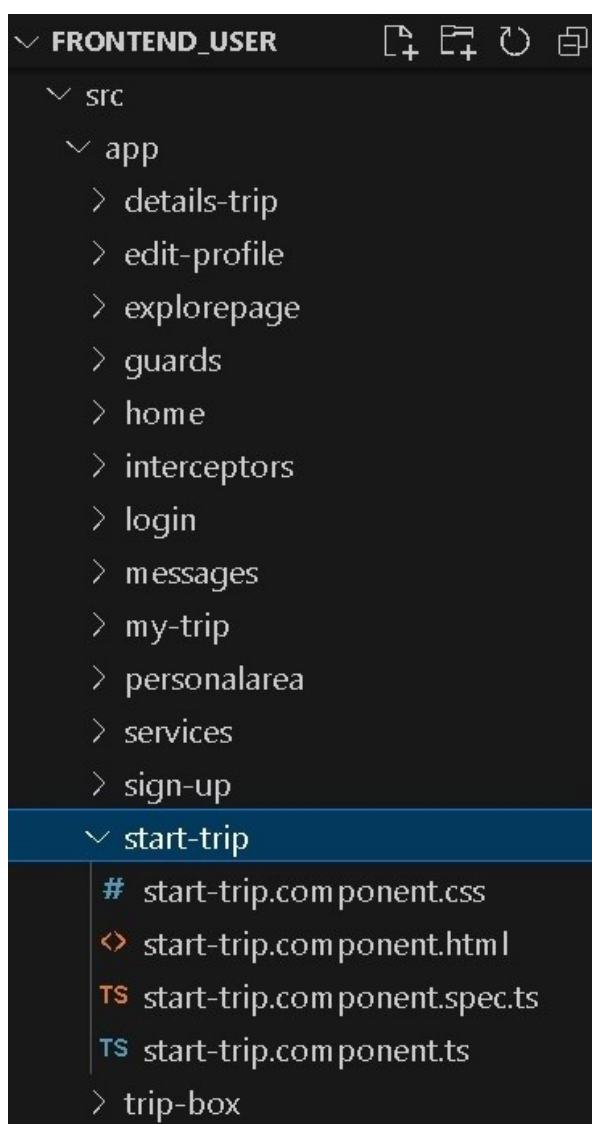


Figura 4.1: Struttura dell'applicazione

4.2 Componenti principali del sistema

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo **Unified Modeling Language (UML)** dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un *tool* per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici ed essenziali.

4.3 Attori

La Figura 4.2 riporta il diagramma rappresentante gli attori.

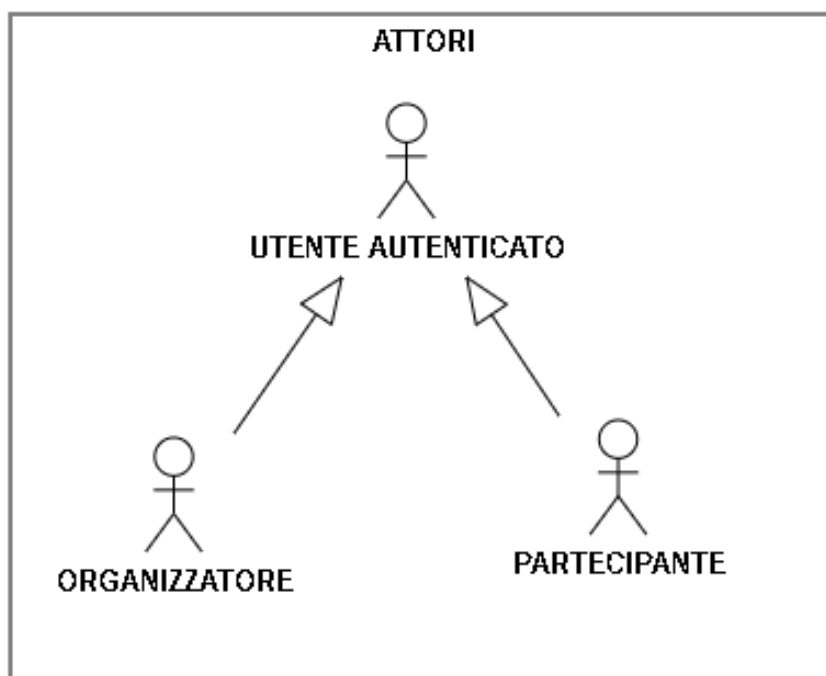


Figura 4.2: Attori

La Figura 4.2 evidenzia la presenza di 3 diversi attori:

- Utente autenticato: qualunque utente che possieda le credenziali per accedere al sito web avendo già effettuato registrazione e login;

Nel diagramma inoltre, si evidenzia una generalizzazione che permette di distinguere un utente autenticato che risulta essere organizzatore o partecipante:

- Organizzatore: è l'utente autenticato che ha creato i parametri per la realizzazione del viaggio;

- Partecipante: è l'utente autenticato che partecipa al viaggio in accordo coi parametri creati dall'organizzatore;

4.4 Use Case

4.4.1 UC01 - Visualizzazione pagina "Esplora Viaggi"

L'utente è già in possesso delle credenziali ed ha già effettuato l'accesso.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e vuole navigare nella pagina "Esplora Viaggi".
- Postcondizioni: l'utente visualizza la pagina "Esplora Viaggi", svolgendo le attività opportune.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Esplora Viaggi"
- l'utente visualizza la pagina "Esplora Viaggi"

4.4.2 UC02 - Visualizzazione pagina "Area Personale"

L'utente è già in possesso delle credenziali ed ha già effettuato l'accesso.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e vuole navigare nella pagina "Area Personale".
- Postcondizioni: l'utente visualizza la pagina "Area Personale", svolgendo le attività opportune.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente visualizza la pagina "Area Personale"

4.4.3 UC03 - Visualizzazione lista viaggi

L'utente vuole visualizzare la lista dei viaggi presenti in "Esplora Viaggi".

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova in "Esplora Viaggi"
- Postcondizioni: l'utente visualizza la lista dei viaggi presenti in "Esplora Viaggi"

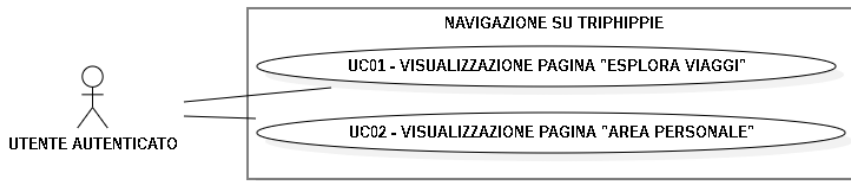


Figura 4.3: Navigazione su TripHippie

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Esplora Viaggi"
- l'utente visualizza la lista dei viaggi

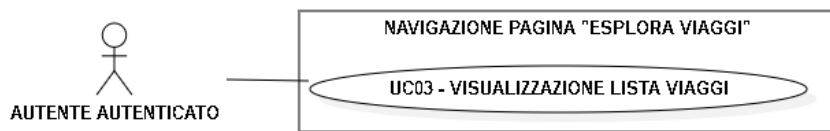


Figura 4.4: UC03

4.4.4 UC03.1 - Visualizzazione singolo viaggio

L'utente vuole visualizzare il singolo viaggio presente nella lista dei viaggi.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova in "Esplora Viaggi"
- Postcondizioni: l'utente visualizza il singolo viaggio della lista dei viaggi presenti in "Esplora Viaggi"

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Esplora Viaggi"
- l'utente visualizza la lista dei viaggi
- l'utente visualizza il singolo viaggio

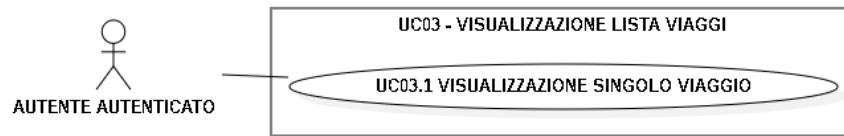


Figura 4.5: UC03.1

4.4.5 UC04 - Modifica viaggio

L'utente vuole modificare il viaggio già precedentemente organizzato.

- Attori primari: Organizzatore
- Precondizioni: l'utente è già loggato, ha già creato un viaggio e si trova nel dettaglio del viaggio creato
- Postcondizioni: l'utente vuole modificare alcuni dati del viaggio

Scenario Principale

- l'utente ha già effettuato il login
- l'utente ha già creato un viaggio
- l'utente visualizza i dettagli del viaggio
- l'utente modifica i dati interessati del viaggio
- l'utente clicca "Modifica"
- l'utente visualizza il messaggio di avvenuta modifica dei dati del viaggio

4.4.6 UC05 - Elimina viaggio

L'utente vuole eliminare il viaggio già precedentemente organizzato.

- Attori primari: Organizzatore
- Precondizioni: l'utente è già loggato, ha già creato un viaggio e si trova nel dettaglio del viaggio creato
- Postcondizioni: l'utente vuole eliminare il viaggio

Scenario Principale

- l'utente ha già effettuato il login
- l'utente ha già creato un viaggio
- l'utente visualizza i dettagli del viaggio
- l'utente clicca "Elimina"
- l'utente visualizza il messaggio di avvenuta eliminazione del viaggio

4.4.7 UC06 - Partecipa

L'utente vuole partecipare ad un viaggio creato da un altro utente.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova nel dettaglio di un viaggio dove non è l'organizzatore
- Postcondizioni: l'utente vuole partecipare al viaggio

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul viaggio desiderato
- l'utente visualizza i dettagli del viaggio
- l'utente clicca "Partecipa"
- l'utente visualizza il messaggio di avvenuta conferma alla partecipazione del viaggio

4.4.8 UC07 - Disiscriviti

L'utente vuole disiscriversi ad un viaggio dove sta partecipando.

- Attori primari: Partecipante
- Precondizioni: l'utente è già loggato e si trova nel dettaglio di un viaggio dove partecipa
- Postcondizioni: l'utente vuole disiscriversi al viaggio

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul viaggio desiderato
- l'utente visualizza i dettagli del viaggio
- l'utente clicca "Disiscrivimi"
- l'utente visualizza il messaggio di avvenuta conferma alla non partecipazione del viaggio

4.4.9 UC08 - Visualizzazione pagina "I miei Viaggi"

L'utente è già loggato e vuole visualizzare i viaggi da lui organizzati e ai quali partecipa.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e vuole visualizzare i propri viaggi.
- Postcondizioni: l'utente visualizza i viaggi da lui organizzati e ai quali partecipa.

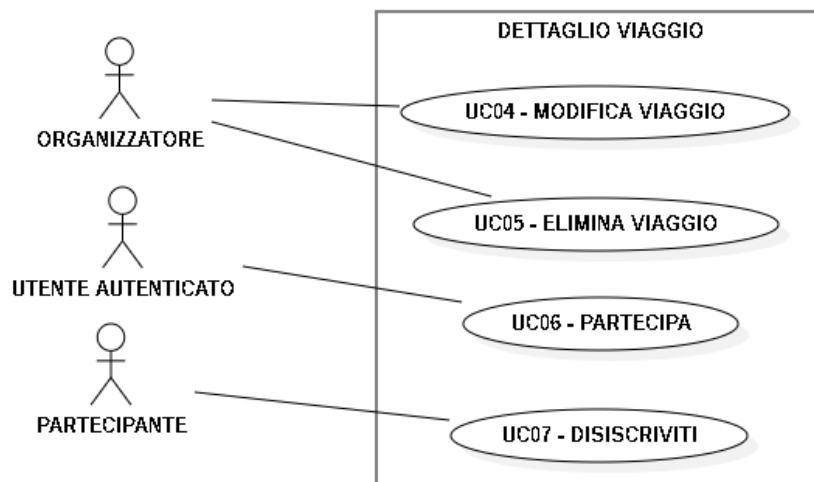


Figura 4.6: Dettaglio viaggio

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "I miei Viaggi"
- l'utente può scegliere se visualizzare i viaggi organizzati o quelli partecipati

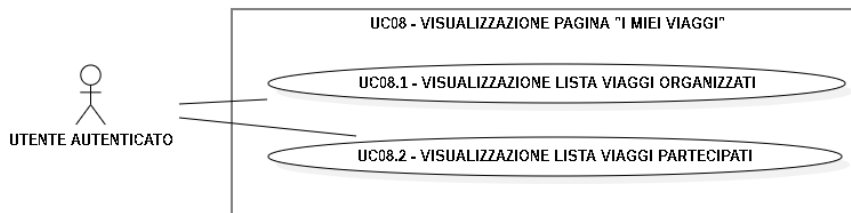


Figura 4.7: UC08

4.4.10 UC08.1 - Visualizzazione lista viaggi organizzati

L'utente è già loggato e vuole visualizzare i viaggi da lui organizzati.

- Attori primari: Organizzatore
- Precondizioni: l'utente è già loggato e si trova in "I miei Viaggi".
- Postcondizioni: l'utente visualizza i viaggi da lui organizzati.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "I miei Viaggi"
- l'utente visualizza i viaggi da lui organizzati

4.4.11 UC08.1.1 - Visualizzazione elemento viaggio organizzato

L'utente è già loggato e vuole visualizzare il singolo viaggio da lui organizzato.

- Attori primari: Organizzatore
- Precondizioni: l'utente è già loggato e si trova in "I miei Viaggi".
- Postcondizioni: l'utente visualizza il singolo viaggio all'interno della lista dei viaggi da lui organizzati.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "I miei Viaggi"
- l'utente visualizza i viaggi da lui organizzati
- l'utente seleziona il singolo viaggio

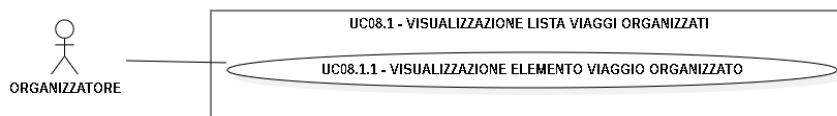


Figura 4.8: UC08.1

4.4.12 UC08.2 - Visualizzazione lista viaggi partecipati

L'utente è già loggato e vuole visualizzare i viaggi a cui partecipa.

- Attori primari: Partecipante
- Precondizioni: l'utente è già loggato e si trova in "I miei Viaggi".
- Postcondizioni: l'utente visualizza i viaggi a cui partecipa.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "I miei Viaggi"
- l'utente clicca sul pulsante "Viaggi Partecipati"
- l'utente visualizza i viaggi a cui partecipa

4.4.13 UC08.2.1 - Visualizzazione elemento viaggio partecipato

L'utente è già loggato e vuole visualizzare il singolo viaggio da lui partecipato.

- Attori primari: Partecipante
- Precondizioni: l'utente è già loggato e si trova in "I miei Viaggi".
- Postcondizioni: l'utente visualizza il singolo viaggio all'interno della lista dei viaggi a cui partecipa.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "I miei Viaggi"
- l'utente clicca sul pulsante "Viaggi Partecipati"
- l'utente visualizza i viaggi a cui partecipa
- l'utente seleziona il singolo viaggio

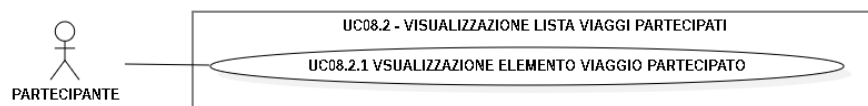


Figura 4.9: UC08.2

4.4.14 UC09 - Visualizzazione pagina "Inizia Viaggio"

L'utente è già loggato e vuole creare un nuovo viaggio.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova in "Inizia Viaggio".
- Postcondizioni: l'utente visualizza i form per la creazione di un nuovo viaggio.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "Inizia viaggio"
- l'utente visualizza la pagina "Inizia Viaggio"

4.4.15 UC10 - Visualizzazione pagina "Modifica Profilo"

L'utente è già loggato e vuole modificare i dati del proprio profilo.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova in "Modifica Profilo".
- Postcondizioni: l'utente visualizza i form per la modifica del proprio profilo.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "Modifica Profilo"
- l'utente visualizza la pagina per modificare i propri dati

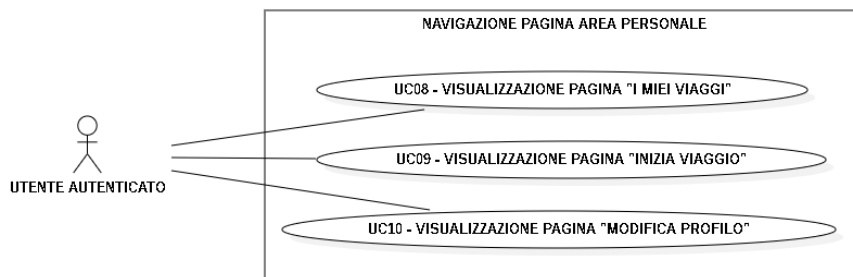


Figura 4.10: Navigazione Area Personale

4.4.16 UC11 - Inserimento campi testuali

L'utente è già loggato e vuole compilare i form di inizio viaggio per la creazione di un nuovo viaggio.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato e si trova in "Inizia Viaggio".
- Postcondizioni: l'utente compila i form per la successiva creazione del viaggio.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "Inizia viaggio"
- l'utente compila i campi testuali

4.4.17 UC12 - Crea nuovo viaggio

L'utente è già loggato e vuole creare un nuovo viaggio.

- Attori primari: Utente autenticato
- Precondizioni: l'utente è già loggato, si trova in "Inizia Viaggio" e ha già compilato i campi testuali.
- Postcondizioni: l'utente crea un nuovo viaggio.

Scenario Principale

- l'utente ha già effettuato il login
- l'utente clicca sul pulsante "Area Personale"
- l'utente clicca sul pulsante "Inizia Viaggio"
- l'utente compila i campi testuali
- l'utente cliccando il pulsante crea un nuovo viaggio del quale ne è organizzatore

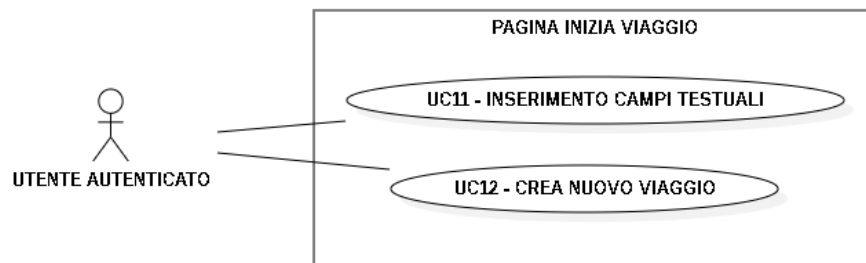


Figura 4.11: Pagina Inizia Viaggio

4.5 Requisiti

4.5.1 Funzionali

Requisito	Descrizione	Fonti
RF1	L'utente deve poter visualizzare la pagina "Esplora Viaggi"	UC01
RF2	L'utente deve poter accedere alla sua area personale	UC02
RF3	L'utente deve poter visualizzare la lista dei viaggi disponibili	UC03
RF4	L'utente deve poter visualizzare i dettagli di un viaggio specifico presente nella lista dei viaggi	UC3.1
RF5	L'organizzatore modifica i dettagli di un viaggio creato	UC04
RF6	L'organizzatore elimina un viaggio da lui creato	UC05
RF7	L'utente partecipa a un viaggio organizzato da un altro utente	UC06
RF8	Il partecipante si disiscrive da un viaggio al quale aveva aderito	UC07
RF9	L'utente visualizza i viaggi da lui organizzati o a cui partecipa	UC08
RF10	L'organizzatore visualizza i viaggi da lui organizzati	UC08.1
RF11	L'organizzatore visualizza i dettagli di un viaggio organizzato	UC08.1.1
RF12	Il partecipante visualizza i viaggi a cui partecipa	UC08.2
RF13	Il partecipante visualizza i dettagli di un viaggio a cui partecipa	UC08.2.1
RF14	L'utente inizia il processo di creazione di un nuovo viaggio	UC09
RF15	L'utente modifica i dettagli del proprio profilo	UC10
RF16	L'utente compila i campi per la creazione di un nuovo viaggio	UC11
RF17	L'utente crea un nuovo viaggio dopo aver compilato i campi richiesti	UC12

Tabella 4.1: Tabella del tracciamento dei requisiti

4.6 Sviluppo Applicazione Web

4.6.1 Home

L'immagine 4.11 rappresenta la pagina di apertura all'interno della quale è possibile effettuare il primo accesso (registrazione) o il login



Figura 4.12: Home

4.6.2 Login

A questa pagina accede solo l'utente già registrato che deve inserire le proprie credenziali per procedere all'autenticazione.

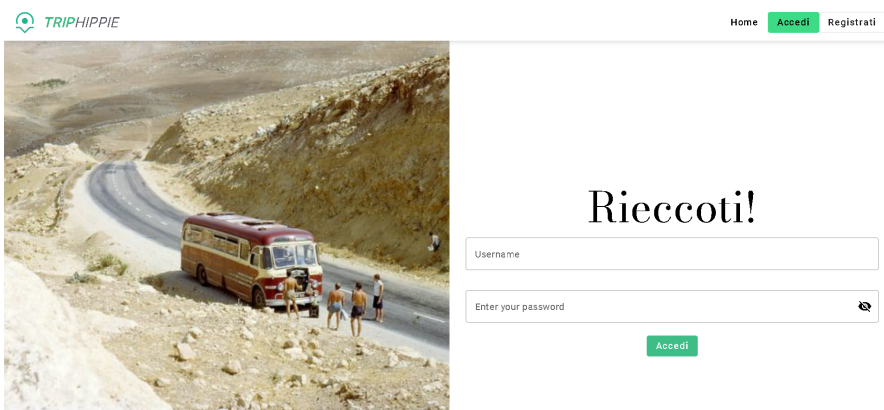


Figura 4.13: Login

4.6.3 Explorepage

In questa sezione è possibile visionare tutti i viaggi già creati dall'utente stesso o da altri utenti. Ogni card permette l'accesso ai dettagli di ogni singolo viaggio. Nell'header è possibile scegliere di tornare all'homepage o di accedere all'area personale.

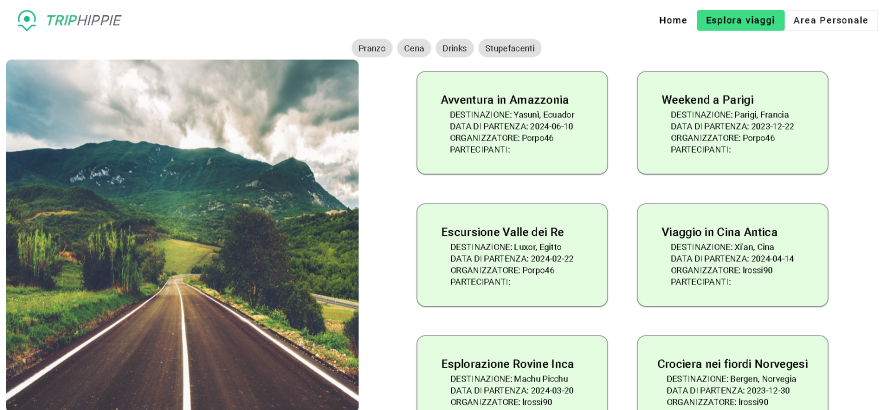


Figura 4.14: Pagina Esplora Viaggi

4.6.4 Details-trip

In questa sezione l'utente può effettuare operazioni diverse a seconda del proprio profilo:

- l'utente autenticato prende visione dei dettagli del viaggio e può eventualmente cliccare il pulsante "Partecipa";
- l'organizzatore può modificare ogni dettaglio del viaggio o eliminare il viaggio stesso;
- il partecipante può prendere visione dei dettagli come l'utente autenticato generico, ma può altresì disiscriversi premendo il pulsante "Disiscriviti".



Figura 4.15: Dettaglio viaggio

4.6.5 My-trip

La sezione si divide in due parti: Viaggi organizzati: si tratta dei viaggi in cui l'utente è l'organizzatore.

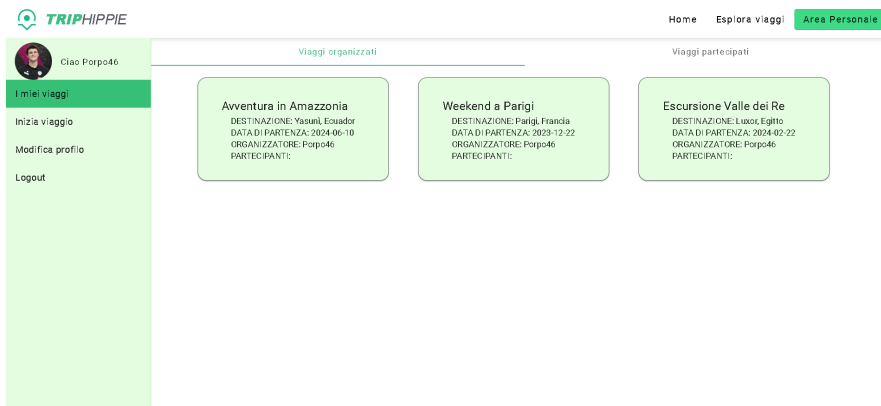


Figura 4.16: Pagina Viaggi Organizzati

Viaggi Partecipati: si tratta dei viaggi in cui l'utente è partecipante.

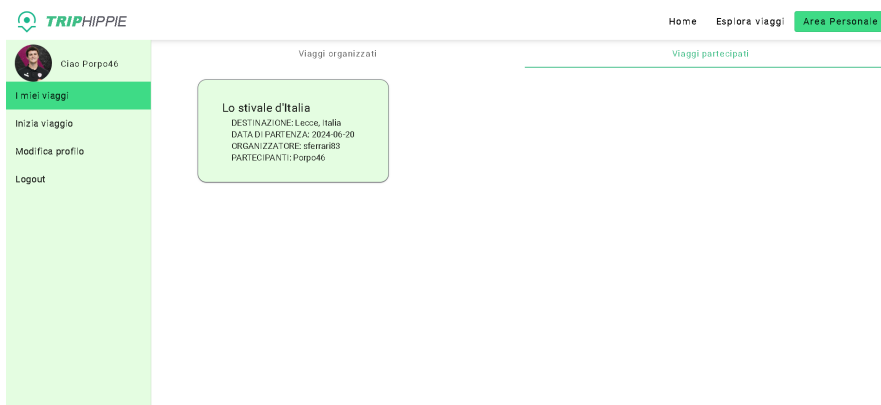


Figura 4.17: Pagina Viaggi Partecipati

4.6.6 Start-trip

Per iniziare il viaggio è obbligatorio compilare tutti i campi di questa sezione:

- Titolo;
- Descrizione;
- Partenza;
- Destinazione;
- Data di partenza;

- Dove vuoi pranzare?;
- Dove vuoi cenare?;

Ulteriori attività tramite checkbox: alcolici e/o stupefacenti. Cliccando il pulsante "Organizza Ora!" si viene reindirizzati alla pagina "Viaggi Organizzati", all'interno della quale apparirà la nuova card relativa al viaggio creato

The screenshot shows the 'Inizia Viaggio' page. At the top, there's a navigation bar with 'Home', 'Esplora viaggi', and 'Area Personale'. The user's profile 'Ciao Porpo46' is visible on the left. The main heading is 'Inizia qui la tua avventura!'. Below it, there are several input fields: 'Titolo', 'Data di partenza' (with a calendar icon), 'Descrizione', 'Dove vuoi pranzare?', 'Partenza', 'Dove vuoi cenare?', and 'Destinazione'. There's also an 'Attività' section with checkboxes for 'Alcolici' and 'Stupefacenti'. A green 'Organizza ora!' button is at the bottom.

Figura 4.18: Pagina Inizia Viaggio

4.6.7 Edit-profile

Questa è la pagina relativa al proprio profilo coi dati personali che è possibile modificare al bisogno.

The screenshot shows the 'Modifica Profilo' page. The navigation bar is the same as in the previous page. The user's profile 'Ciao Porpo46' is visible on the left. The main heading is 'Modifica immagine profilo'. Below it, there are several input fields: 'Nome' (value: Davide), 'Cognome' (value: Porparati), 'Data di nascita' (value: 11/27/1999), 'Username' (value: Porpo46), 'Email' (value: davide.porparati@studenti.unipd.it), 'Nuova Password', and 'Conferma nuova password'. There are also 'Modifica dati' and 'Elimina account' buttons at the bottom.

Figura 4.19: Pagina Modifica Profilo

Capitolo 5

Conclusioni

In questo capitolo si dà un consuntivo delle attività, viene analizzato il raggiungimento degli obiettivi prefissati ad inizio stage e vengono espone delle riflessioni sul percorso di stage.

5.1 Raggiungimento degli obiettivi

Gli obiettivi prefissati a inizio stage risultano totalmente raggiunti. Come mostrato nella Tabella 5.1, tutti gli obiettivi sono stati raggiunti.

Obbligatorio		
OB1	Acquisizione competenze sulle tematiche sopra descritte	SI
OB2	Capacità di raggiungere gli obiettivi richiesti in autonomia seguendo il cronoprogramma	SI
OB3	Portare a termine le implementazioni previste con una percentuale di superamento pari al 80% (equivalente alla maschera di login/registrazione ed inserimento Viaggio e loro service con le chiamate al <i>backend</i>)	SI
Desiderabile		
DE1	Portare a termine le implementazioni previste con una percentuale di superamento pari al 100% (equivalente alla maschera di login/registrazione, Inserimento, gestione e visualizzazione Viaggi e loro service con le chiamate al <i>backend</i>)	SI
Opzionale		
OP1	Apportare un valore aggiunto al gruppo di lavoro durante le fasi di progettazione delle interfacce	SI

Tabella 5.1: Tabella degli obiettivi raggiunti

5.2 Punti di forza e di debolezza del progetto

Il mio percorso di stage si è rivelato un'ottima occasione per ampliare notevolmente le mie competenze nel contesto delle tecnologie utilizzate, specialmente nello sviluppo front-end di applicazioni web. Nel corso delle prime fasi del periodo di studio, ho acquisito una solida conoscenza di Angular, il che ha agevolato il mio successivo impegno nello sviluppo del codice. Le riunioni di [Stato Avanzamento Lavori \(SAL\)](#), che avevano luogo con cadenza settimanale, cui prendevano parte tutti gli stagisti facenti parte del progetto TripHippie insieme all'ing. Pallaro e al dott. Zorzi, si sono rivelate un utile momento di scambio di informazioni, consigli, suggerimenti, per progettare e programmare obiettivi comuni da raggiungere. Dopo quest'esperienza posso certo fare mia la citazione di *R.J. Meehan*¹ che dichiara: "La risorsa più preziosa che tutti hanno sono i colleghi. Senza collaborazione la nostra crescita personale è limitata alle nostre prospettive". Tuttavia, mi sono trovato nella condizione di dover affrontare alcune sfide durante le prime fasi del processo di sviluppo. Solo per citare un esempio, ci si è trovati nella condizione di lavorare sul codice back-end e front-end con tempistiche differenti. Il gruppo di cui facevo parte ha operato sul front-end a partire da settembre, mentre il gruppo incaricato del back-end aveva concluso la sua attività già a fine agosto. Ciò ha comportato quindi la necessità di modificare chiamate di back-end in funzione di quello che si voleva ottenere nella parte di front-end. Tali operazioni hanno a volte influito sui tempi di realizzazione del mio percorso. Nonostante le difficoltà incontrate, ho considerato queste sfide come opportunità per approfondire la mia comprensione delle tecnologie coinvolte e ho acquisito una serie di conoscenze preziose che non avrei potuto ottenere altrimenti.

5.3 Considerazioni finali

Lo stage è stato un'esperienza estremamente positiva, per l'acquisizione di una visione più chiara del mondo del lavoro nel settore informatico. La possibilità di lavorare su un progetto che avevo scelto personalmente, affrontando argomenti che mi appassionano, ha contribuito in modo significativo al mio apprezzamento di questa attività. Ho trovato particolarmente stimolante la possibilità di approfondire nuovi argomenti esclusivamente attraverso la consultazione di documentazione online, in contrasto con l'approccio più tradizionale seguito durante il mio percorso di studi. Inoltre, l'interazione con colleghi stagisti e con i numerosi dipendenti di SyncLab è stata un'ulteriore occasione di crescita professionale in un ambiente aziendale specifico del settore. Ho sempre cercato, nelle varie fasi del mio lavoro, di portare il mio contributo personale e, al contempo, di imparare dalle esperienze altrui e dai contesti diversi in cui ero inserito, con estrema umiltà e apertura all'ascolto, il che mi ha permesso di trovare punti di contatto con le persone con cui ho collaborato. Ritengo quindi che questo stage abbia rappresentato una componente fondamentale del mio percorso di formazione, così come della mia crescita motivazionale. È stato altresì un utile momento di riflessione, nonché un ottimo punto di partenza per la mia futura attività professionale.

¹John Meehan - commediografo, sceneggiatore, regista canadese (1890-1954).

Acronimi e abbreviazioni

API REST Representational State Transfer (REST) API. 13, 37

CSS Cascading Style Sheets. 10, 37

HTML Hyper Text Markup Language. 9, 37

IDE Integrated Development Environment. 13, 37

MVC Model-View-Controller. 17, 39

MVVM Model-View-ViewModel. 17, 39

SAL Stato Avanzamento Lavori. 36, 37

SQL Structured Query Language. 15, 37

UML Unified Modeling Language. 19, 39

VSC Visual Studio Code. 13, 37

Glossario

Car Sharing Il Car Sharing è un servizio di mobilità che offre agli utenti l'opportunità di noleggiare veicoli, solitamente automobili, per un periodo di tempo limitato, spesso in base all'uso orario o giornaliero.. [3](#), [39](#)

Component In Angular è un modulo riutilizzabile che integra logica e interfaccia utente. Essenziale per la costruzione di applicazioni web modulari.. [10](#), [39](#)

Framework Un framework è uno scheletro su cui costruire software, fornendo una struttura e delle regole predefinite. Semplifica quindi lo sviluppo fornendo un set di strumenti e convenzioni.. [10](#), [39](#)

MVC L'architettura MVC è un design pattern che guida la creazione di un'applicazione software suddivisa in tre componenti principali: il Model, la View e il Controller.. [37](#)

MVVM L'architettura MVVM è un design pattern che conduce alla creazione di un'applicazione software divisa in tre componenti principali: il Model, il ViewModel e la View.. [37](#)

Software House Una software house è essenzialmente una società specializzata nello sviluppo di software. Queste aziende creano applicazioni, programmi e sistemi informatici per soddisfare le esigenze dei loro clienti. Possono occuparsi di progettazione, sviluppo, testing e manutenzione del software, offrendo soluzioni personalizzate per diverse industrie e settori. . [1](#), [39](#)

Typescript Linguaggio di programmazione che implementa "JavaScript" aggiungendo la tipizzazione. [9](#), [39](#)

UML in ingegneria del software *UML*, *Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. [37](#)

Sitografia

Siti web consultati

Componenti Angular:

URL: <https://dariopironi.com/it/componenti-angular-cosa-sono-e-come-crearli>.

Diagrammi dei casi d'uso:

URL: https://it.wikipedia.org/wiki/Use_Case_Diagram.

Documentazione Angular:

URL: <https://angular.io/docs>.

Documentazione CSS:

URL: <https://www.w3schools.com/css/>.

Documentazione HTML:

URL: <https://www.w3schools.com/html/>.

Documentazione JAVA:

URL: <https://www.oracle.com/java/>.

Documentazione MVC Pattern Angular:

URL: <https://medium.com/@maaouikimo/why-angularis-your-best-choice-for-you-next-projects-9d754fb18f91>.

Documentazione MVVM Pattern :

URL: <https://it.wikipedia.org/wiki/Model-view-viewmodel>.

Documentazione Spring:

URL: <https://docs.spring.io/spring-framework/reference/overview.html>.

Documentazione SpringBoot:

URL: <https://spring.io/projects/spring-boot>.

Documentazione Swagger:

URL: <https://www.geekandjob.com/wiki/swagger>.

Documentazione Typescript:

URL: <https://www.typescriptlang.org/docs/home.html>.

Figma :

URL: <https://www.pixartprinting.it/blog/figma-cos-e/>.

Framework Angular:

URL: <https://angular.io/>.

Introduzione a Spring:

URL: <https://www.youtube.com/watch?v=wBRpPhGkUjo>.

Synclab:

URL: <https://www.synclab.it/>.

Visual Studio Code:

URL: <https://visualstudio.microsoft.com/it/>.