

REPORT TAW

ARCHITETTURA DEL SISTEMA

Il sistema è suddiviso in tre macro-componenti, ciascuna eseguita in un container Docker separato:

- **Frontend SPA:** sviluppata in Angular, si occupa dell'interfaccia utente e della gestione delle interazioni lato client. Comunica con il backend tramite REST API.
- **Backend REST API:** sviluppato con Node.js e utilizza il framework Express.js per gestire le rotte e le richieste. Fornisce tutte le funzionalità richieste dall'applicazione tramite un set di API REST, tra cui la registrazione e autenticazione degli utenti, la gestione di rotte e voli da parte delle compagnie aeree, l'acquisto dei biglietti, la selezione dei posti e la visualizzazione delle statistiche. Il backend si occupa anche della validazione dei dati e della gestione dei ruoli (admin, compagnia aerea, utente normale).
- **Database:** L'applicazione utilizza un database relazionale PostgreSQL per memorizzare tutte le informazioni necessarie, come utenti, compagnie, voli, rotte, biglietti e aerei. All'avvio del backend, viene effettuato un pre-caricamento di dati di test, utile per simulare il funzionamento del sistema. Il database è accessibile solo dal backend e non è esposto direttamente all'esterno.

Tutti e tre i componenti sono **containerizzati con Docker** e gestiti tramite **Docker Compose**, che consente di avviare l'intera applicazione con un solo comando. I container comunicano tra loro all'interno della stessa rete Docker. In particolare:

- il **frontend** invia richieste al backend,
- il **backend** interroga il database,
- il **database** risponde alle query del backend.

DATABASE

Il progetto utilizza un database relazionale basato su PostgreSQL, modellato attraverso Prisma ORM. Il modello dati è stato progettato per rappresentare in maniera chiara e normalizzata le entità principali di un sistema di gestione voli, compagnie aeree e prenotazioni.

Di seguito viene descritta ogni entità del modello:

- **Users:** rappresenta un utente registrato al sistema. Ogni utente ha un nome, un'e-mail univoca, una password e un ruolo (ad esempio utente o amministratore).
- **Trips:** un viaggio associato a un singolo utente. Serve a raggruppare fino a due prenotazioni legate a una stessa esperienza di viaggio (ad esempio andata e ritorno).
- **Airports:** rappresenta un aeroporto identificato da nome, città, paese e coordinate geografiche. Utilizzato per definire le rotte.
- **Routes:** definisce una tratta tra un aeroporto di partenza e uno di destinazione.
- **Airlines:** una compagnia aerea registrata. Ogni compagnia ha un nome (chiave primaria), password, paese e un motto.
- **Aircrafts:** rappresenta un aereo, con modello e capacità di posti. È di proprietà di una compagnia aerea.
- **Seats:** ogni posto è legato a un aereo specifico. Ha una posizione identificativa (es. "A1").
- **Flights:** un volo associato a una compagnia aerea, a un aereo, a una rotta e a una data/orario di partenza.
- **Tickets:** rappresenta l'offerta di un volo per un certo prezzo e una certa classe (es. economy, business).
- **Bookings:** una prenotazione effettuata da un utente per un viaggio. Include riferimento al posto scelto, al biglietto e a eventuali extra.
- **Extras:** rappresenta un servizio aggiuntivo acquistabile con la prenotazione
- **Uses:** tabella che rappresenta l'associazione tra compagnie aeree e rotte che esse utilizzano.
- **BUWTs:** tabella che memorizza i token JWT invalidati, usata per la gestione della sicurezza e del logout.

Tutte le entità sono modellate nel file schema.prisma, dove sono anche definite le relazioni tra tabelle tramite chiavi esterne, con le relative opzioni onDelete e onUpdate per mantenere la consistenza referenziale.

API REST

Il backend del sistema TAW è realizzato con Node.js e il framework Express, esponendo un'API RESTful per comunicare con il frontend Angular.

La configurazione del server avviene tramite alcune variabili d'ambiente:

- **DATABASE_URL:** URL del database PostgreSQL, usato da Prisma per la connessione
- **JWT_SECRET:** chiave segreta per la firma e verifica dei JSON Web Token (JWT)
- **JWT_EXPIRATION:** durata di validità dei token JWT (es. 1h)
- **PORT:** porta su cui gira il server (default 3000)

Il backend è organizzato in moduli per gestire utenti, compagnie aeree, navigazione e prenotazioni, con middleware dedicati per autenticazione, validazione dati e controllo autorizzazioni.

Account (users)

Metodo	Endpoint	Descrizione	Parametri/Body
POST	/api/users/user	Creazione nuovo utente	JSON body con dati utente (es. email, password, nome)
GET	/api/users/user	Ottiene dati utente autenticato	Header: Authorization con token JWT
POST	/api/users/login	Login utente	JSON body con email e password
DELETE	/api/users/user	Cancellazione utente autenticato	Header: Authorization con token JWT
GET	/api/users/accounts	Lista utenti (solo admin)	Header: Authorization con token JWT
DELETE	/api/users/accounts/:id	Rimozione utente per ID (solo admin)	Header: Authorization con token JWT

Compagnie aeree (Airlines)

Metodo	Endpoint	Descrizione	Parametri/Body
POST	/api/airlines/invite	Invito per nuova compagnia	JSON body con dati invitante
POST	/api/airlines/enroll/:invitationCode/:airlineName	Registrazione compagnia tramite codice invito	Parametri path: invitationCode, airlineName
POST	/api/airlines/login	Login compagnia aerea	JSON body con credenziali
GET	/api/airlines/airlines	Lista compagnie aeree	Header: Authorization con token JWT
POST	/api/airlines/aircrafts	Aggiunta nuovo aereo	JSON body con dati aereo
GET	/api/airlines/aircrafts/:airlineName	Lista aerei per compagnia	Parametro path: airlineName
DELETE	/api/airlines/aircrafts/:aircraftId	Rimozione aereo	Parametro path: aircraftId
POST	/api/airlines/routes	Creazione nuova rotta	JSON body con dati rotta
DELETE	/api/airlines/routes/:routeId	Cancellazione rotta	Parametro path: routeId
GET	/api/airlines/routes	Lista rotte	Header: Authorization con token JWT
GET	/api/airlines/flights	Lista voli	Header: Authorization con token JWT

Metodo	Endpoint	Descrizione	Parametri/Body
GET	/api/airlines/flights/pending	Lista voli in attesa di approvazione	Header: Authorization con token JWT
POST	/api/airlines/tickets	Creazione biglietti	JSON body con dati biglietto

Navigazione (Navigation)

Metodo	Endpoint	Descrizione	Parametri
GET	/api/navigate/airports	Lista aeroporti	Nessuno
GET	/api/navigate/airports/:departure/routes	Rotte in partenza dall'aeroporto specificato	Parametro path: departure
GET	/api/navigate/airports/:destination/incoming-routes	Rotte in arrivo all'aeroporto specificato	Parametro path: destination
GET	/api/navigate/routes/path?from=1&to=16	Ricerca percorso multi-tratta tra aeroporti	Query params: from, to
POST	/api/navigate/routes	Creazione nuova rotta	JSON body con dati rotta
POST	/api/navigate/flights	Creazione nuovo volo	JSON body con dati volo
GET	/api/navigate/flights	Lista voli	Nessuno

Prenotazioni (Bookings)

Metodo	Endpoint	Descrizione	Parametri
GET	/api/bookings/seats/:flightUUID	Lista posti disponibili per volo	Parametro path: flightUUID
GET	/api/bookings/tickets/:flightUUID	Lista biglietti per volo	Parametro path: flightUUID
POST	/api/bookings/trips	Creazione nuovo viaggio	JSON body con dati viaggio
GET	/api/bookings/trips/:tripId	Dettagli viaggio	Parametro path: tripId

AUTENTICAZIONE

L'autenticazione degli utenti nel sistema è implementata tramite l'uso di JWT (JSON Web Token), una soluzione che permette sia di autenticare sia di autorizzare gli utenti in modo sicuro e scalabile.

Meccanismo di autenticazione:

- Al momento del login o della registrazione, l'utente (o la compagnia aerea) invia le proprie credenziali tramite un apposito endpoint REST.
- Se le credenziali sono valide, il backend genera un token JWT che include informazioni utili per l'identificazione e il controllo dei permessi, come ad esempio il tipo di utente (utente normale, amministratore o compagnia aerea) e, nel caso delle compagnie, il nome associato.
- Questo token viene restituito al frontend e salvato nel local storage del browser, per permettere l'accesso alle risorse protette nelle successive richieste HTTP.

Sicurezza e autorizzazione:

- Il token JWT ha una durata limitata, impostata per ridurre il rischio di utilizzo fraudolento in caso di furto del token.
- Ogni richiesta verso endpoint protetti deve includere il token JWT nell'header Authorization come Bearer token.
- Sul backend, tramite middleware dedicati, ogni token viene validato e decodificato per:
 - Verificare l'autenticità e la scadenza.
 - Estrarre le informazioni di ruolo e permessi.
 - Consentire o negare l'accesso alle risorse in base ai privilegi dell'utente.

Workflow di autenticazione tipico:

1. L'utente invia credenziali di login (username/e-mail + password) all'endpoint `/api/users/login`.
2. Il backend verifica le credenziali e, se corrette, genera e restituisce un JWT.
3. Il frontend memorizza il token JWT nel local storage.
4. Per ogni richiesta successiva a risorse protette, il frontend include il token nell'header Authorization.

5. Il backend, tramite middleware, verifica il token e, se valido, permette l'accesso alla risorsa richiesta.
6. Se il token è scaduto o non valido, il backend risponde con un errore di autenticazione, che può portare il frontend a richiedere un nuovo login.

ANGULAR FRONTEND

L'interfaccia utente è sviluppata come Single Page Application (SPA) utilizzando il framework Angular (versione 19.2.13). Il progetto è organizzato in maniera modulare, con componenti riutilizzabili e pagine logiche che gestiscono le varie funzionalità per gli utenti (admin, compagnie aeree, clienti).

Struttura principale

1) Components: contiene i componenti riutilizzabili, utilizzati per visualizzare elementi specifici dell'interfaccia utente. Tra i principali:

- **aircraft-card, airline-card, airline-flight-card, airport-card, flight-card, route-card, trip-card, user-card:** mostrano rispettivamente le informazioni e permettono modifiche riguardanti aircrafts (visibile da airline), airlines (visibile da admin), voli creati (visibile da airline, con possibile eliminazione), aeroporti (visibile da admin), voli disponibili (visibile clienti, airlines e admin), rotte (visibile da airline), viaggio (visibile da clienti), utenti (visibile da admin, con possibile eliminazione).
- **flight-list, flight-search:** lista e gestione della ricerca dei voli.
- **Map, map-routing:** integrazione con mappa (Leaflet) rispettivamente nella homepage e nella lista dei voli.
- **toolbar:** barra di navigazione principale presente in tutte le pagine.

2) pages: raccoglie i componenti di alto livello che rappresentano vere e proprie pagine dell'applicazione. I principali moduli includono:

- **admin-dashboard:** con sottosezioni airline-registration, airlines-management, airports-management, users-management.
- **airline-enrollment:** permette l'accesso al sito e il cambio credenziali di una airline tramite invito.
- **airline-login:** permette il login di una airline dopo aver eseguito l'enrollment.
- **flights-display:** mostra l'elenco di tutti i voli trovati in base ai dati inseriti nel componente flight-search.
- **forbidden:** mostra un messaggio di "Accesso negato" quando un utente tenta di accedere a una risorsa o pagina per cui non ha i permessi necessari.
- **homepage:** è la pagina iniziale del sito, contiene una mappa con tutti gli aeroporti del mondo, permette la ricerca dei voli tramite il componente flight-search e contiene la toolbar con cui è possibile venire indirizzati alla pagina di login o le pagine specifiche per gli utenti e le airlines in base al ruolo con cui si è autenticati.

- **homepage-airline:** con sottosezioni aircrafts-management, flights, management, routes-management.
- **profiles:** permette la visualizzazione e il download dei voli prenotati e le relative informazioni.
- **server-error:** pagina utilizzata principalmente a scopo di test dell'applicazione e permette di capire quando il server non è in funzione.
- **signin:** tramite indirizzo e-mail e password è possibile eseguire il signin al sito.
- **ticket-booking:** mostra un riepilogo del volo scelto e permette di confermare la prenotazione del volo
- **user-registration:** permette di effettuare la registrazione di un cliente al sito inserendo nome, indirizzo e-mail e password.

3) Guards: Contiene le route guard per proteggere le sezioni riservate in base al ruolo dell'utente autenticato tramite JWT.

In particolare, con l'implementazione attuale protegge le rotte admin-dashboard e homepage-airline.

4) Interceptors: Gli intercettori servono per gestire automaticamente l'inserimento del token JWT e gli errori di comunicazione con il server, migliorando la sicurezza e l'esperienza utente.

5) Environments: Contiene la variabile d'ambiente apiUrl: 'http://129.152.7.218:3000' con l'indirizzo del server che verrà utilizzata nei vari endpoints per le richieste API.

Routing

Il file app.routes.ts definisce le principali rotte dell'applicazione, associando ogni percorso a un componente e utilizzando canActivate e RoleAuthGuard per limitare l'accesso in base al ruolo.

- **Admin** (ruolo 1): accesso a admin-dashboard, airline-registration, users-management, airports-management, ecc.
- **Airline** (ruolo 2): accesso a homepage-airline, flights-management, aircrafts-management, routes-management, ecc.
- **Cliente** (ruolo 0): accesso a homepage, profile-customer, flight-display, ticket-booking.

- **/** -> homepage
- **/signin** -> login
- **/forbidden** -> accesso negato
- **/admin-dashboard** -> dashboard amministratore (solo ruolo 1)
 - **/airline-registration** -> registrazione airline
 - **/airlines-management** -> gestione airlines
 - **/airports-management** -> gestione aeroporti
 - **/users-management** -> gestione aeroporti
- **/airline-enrollment**
 - **/forbidden**
- **/airline-enrollment/:invitationCode/:airlineName** -> iscrizione compagnia aerea tramite invito
- **/airline-login** -> login per airlines
- **/user-registration** -> registrazione nuovo cliente
- **/homepage-airline** -> dashboard airline (solo ruolo 2)
 - **/routes-management** -> gestione rotte
 - **/aircrafts-management** -> gestione aerei
 - **/flights-management** -> gestione voli
- **/server-error** -> errore del server
- **/flights-display** -> voli disponibili
- **/profile-customer** -> profile cliente con viaggi e biglietti
- **/ticket-booking** -> prenotazione biglietti (solo ruolo 0)

Interfaccia utente

È stata utilizzata principalmente la libreria angular material per fornire componenti grafici moderni e accessibili, ad esempio sono stati utilizzati MatButtonModule, MatCardModule, FormsModule.

ESEMPI

Benvenuto su TAW Flights!

Cerca voli tra centinaia di destinazioni. Acquista biglietti e gestisci le tue prenotazioni.

Tipo viaggio*

Solo andata

Da*

A*

Data di partenza*

Viaggiatori*

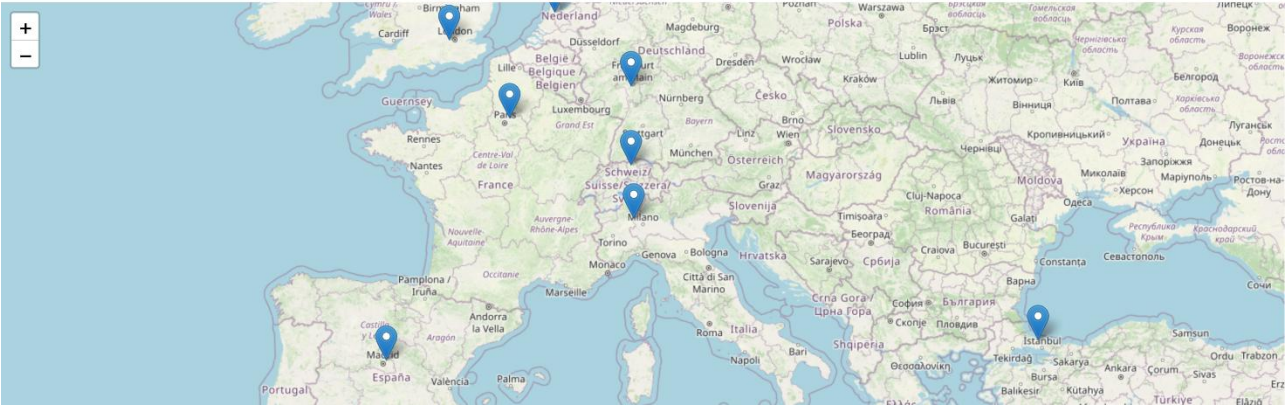
1

Classe*

ECONOMY

☐ Solo voli diretti

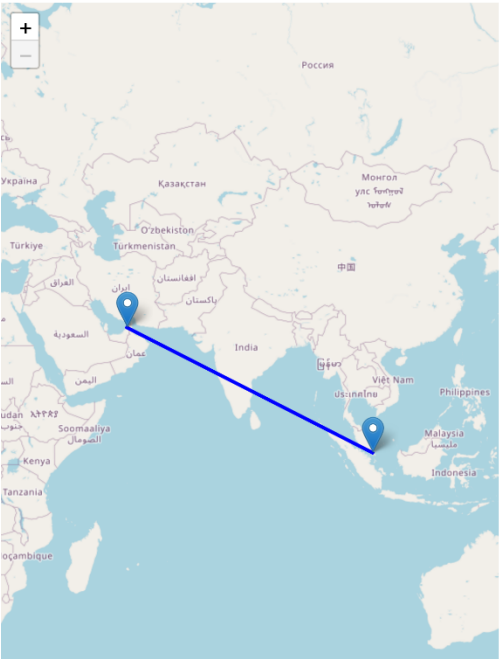
Cerca voli



TAW Flights

[Accedi al sistema](#)

Risultati ricerca voli



Singapore Airlines

11:00
Dubai International
(Dubai)

6h 30m
Diretto

16:30
Changi Airport
(Singapore)

€264

Accedi al sistema

Email*

Password*

Login

Non sei registrato? [Registrati](#)

Sei una compagnia aerea? [Login compagnia aerea](#)

- Invita Compagnia
- Gestione Compagnie
- Gestione Utenti
- Gestione Aeroporti

Gestione Aeroporti

Ordina per: Nome

Nome:	Città:	Nazione:	Fuso orario:
Amsterdam Schiphol	Amsterdam	Netherlands	UTCEurope/Amsterdam
Beijing Capital	Beijing	China	UTCAsia/Shanghai
Changi Airport	Singapore	Singapore	UTCAsia/Singapore
Charles de Gaulle	Paris	France	UTCEurope/Paris
Dubai International	Dubai	United Arab Emirates	UTCAsia/Dubai
Frankfurt Airport	Frankfurt	Germany	UTCEurope/Berlin
Heathrow	London	UK	UTCEurope/London
Istanbul Airport	Istanbul	Turkey	UTCEurope/Istanbul

TAW Flights

Admin

Invita Compagnia

Gestione Compagnie

Gestione Utenti

Gestione Aeroporti

Dashboard Amministratore

Gestione Utenti

Nome: John Doe

Email: john@email.com

Password:

Nome: Admin User

Email: admin@airline.com

Password:

Nome: Mario

Email: mario@gmail.com

Password:

Nome: Admin

Email: adminAccount@gmail.com

Password:

TAW Flights

Admin

Invita Compagnia

Gestione Compagnie

Gestione Utenti

Gestione Aeroporti

Dashboard Amministratore

Invita una nuova compagnia aerea

Solo un amministratore può invitare una nuova compagnia aerea. Inserisci il nome della compagnia per generare un invito.

Nome compagnia aerea*

Invia invito

[← Torna al login](#)

TAW Flights

Admin

Dashboard Amministratore

Invita Compagnia

Gestione Compagnie

Gestione Utenti

Gestione Aeroporti

Gestione Compagnie Aeree

Nome:

FlySafe

Paese:

Francia

Motto:

Vive la france

Nome:

Emirates

Paese:

United Arab Emirates

Motto:

Fly Better

Nome:

Lufthansa

Paese:

Germany

Motto:

Say yes to the world

Nome:

Singapore Airlines

Paese:

Singapore

Motto:

A Great Way to Fly

TAW Flights

FlySafe

Dashboard Compagnia

Gestione Rotte

Gestione Aerei

Gestione Voli

Gestione Aerei

+ Aggiungi aereo

ID:5

Modello:Boeing 747-8

Posti:320

ID:6

Modello:Boeing 777

Posti:300

ID:7

Modello:Airbus A320

Posti:180

ID:8

Modello:Airbus A321

Posti:200

TAW Flights

FlySafe

Compagnia Aerea

Dashboard Compagnia

Gestione Rotte

Gestione Aerei

Gestione Voli

Gestione Voli

+ Aggiungi volo

Lista voli

Codice: 738ce3e9-8716-4ec7-a2f3-e2e2453b8a74

Boeing 747-8

Luogo di partenza: Dubai International (Dubai, United Arab Emirates)

Destinazione: Frankfurt Airport (Frankfurt, Germany)

Partenza: 2025-06-25T10:00:00

Durata: 240 min

Posti: 320

Fuso orario: Asia/Dubai

Modello aereo: Boeing 747-8

Capacità posti: 320

Codice: 205ef1db-3537-4bc0-9487-355d4cafaf3b

Boeing 777

Luogo di partenza: Frankfurt Airport (Frankfurt, Germany)

Destinazione: Dubai International (Dubai, United Arab Emirates)

Partenza: 2025-06-26T10:00:00

Durata: 600 min

Posti: 300

Fuso orario: Europe/Berlin

Modello aereo: Boeing 777

Capacità posti: 300

Codice: e2d106eb-de7e-4846-b377-a6242d182974

Airbus A320

Luogo di partenza: Changi Airport (Singapore, Singapore)

Destinazione: Tokyo Haneda (Tokyo, Japan)

TAW Flights

Mario

I tuoi viaggi

Viaggio #1

Prenotazione 1

Data partenza: 6/25/25, 7:00 AM

Da: Dubai International

A: John F. Kennedy International

Codice volo: 85e20c25-a8ee-4f7b-88c5-d9ae48454d7d

Classe: ECONOMY

Scarica dettagli

Stato viaggio: Attivo

Viaggio #2

Prenotazione 1

Data partenza: 6/25/25, 8:00 AM

Da: Dubai International

A: Frankfurt Airport

Codice volo: 738ce3e9-8716-4ec7-a2f3-e2e2453b8a74

Classe: BUSINESS

Scarica dettagli

Prenotazione 2

Data partenza: 6/24/25, 10:00 PM

Da: Frankfurt Airport

A: Toronto Pearson

Codice volo: c39a230f-5f7f-4b76-a68e-8f22caf0b099

Classe: DELUXE

Scarica dettagli

Stato viaggio: Attivo

Viaggio #3