

Daniele Guardascione

Lubrano Vincenzo Pio

N86005497

N86005417



NOME PROGETTO: SOFTWARE PER LA GESTIONE
DELL'AEROPORTO DI NAPOLI

Traccia 3: Aeroporto

Si sviluppi un sistema informativo per la gestione dell'aeroporto di Napoli, composto da una base di dati relazionale e da un applicativo Java con interfaccia grafica

realizzata con Swing. Questo sistema deve consentire di organizzare e monitorare le operazioni aeroportuali in modo efficiente e intuitivo.

Il sistema può essere utilizzato da utenti autenticati tramite una login e una password. Gli utenti sono suddivisi in due ruoli: utenti generici, che possono prenotare voli, e amministratori del sistema, che gestiscono l'inserimento e l'aggiornamento dei voli.

Il sistema gestisce i voli in arrivo e quelli in partenza. Ogni volo è caratterizzato da un codice univoco, la compagnia aerea, l'aeroporto di origine (per i voli in arrivo a Napoli) e quello di destinazione (per i voli in partenza da Napoli), la data del volo, l'orario previsto, l'eventuale ritardo e lo stato del volo (programmato, decollato, in ritardo, atterrato, cancellato). Gli amministratori del sistema hanno la possibilità di inserire nuovi voli e aggiornare le informazioni sui voli esistenti.

Gli utenti generici possono effettuare prenotazioni per i voli programmati. Ogni prenotazione è legata a un volo e contiene informazioni come i dati del passeggero (che non deve necessariamente coincidere con il nome dell'utente che lo ha prenotato), il numero del biglietto, il posto assegnato e lo stato della prenotazione (confermata, in attesa, cancellata). Gli utenti possono cercare e modificare le proprie prenotazioni in base al nome del passeggero o al numero del volo.

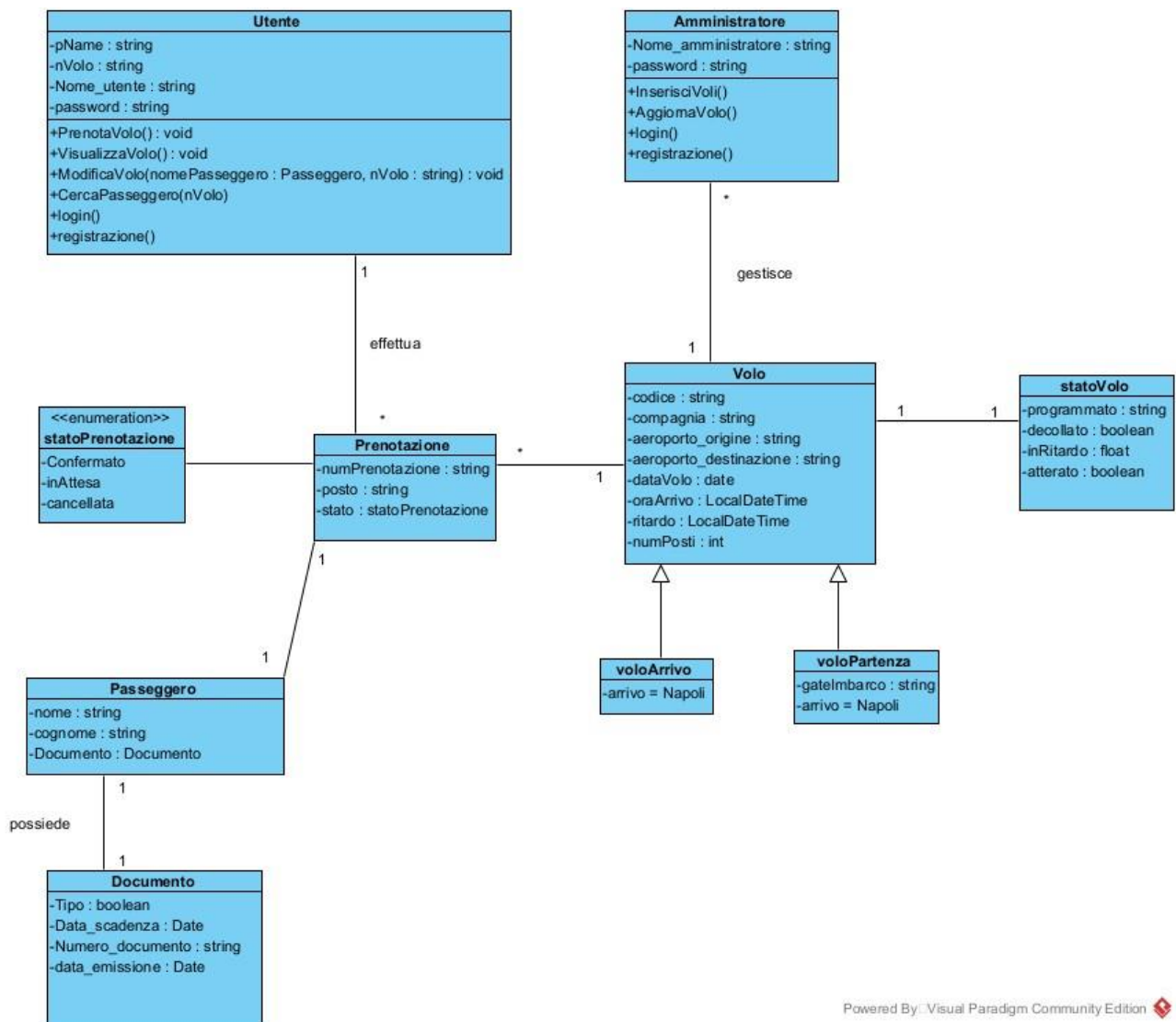
Il sistema gestisce anche i gate di imbarco (identificati da un numero), assegnandoli

ai voli in partenza. Gli amministratori possono modificare l'assegnazione dei gate.

Il sistema consente agli utenti di visualizzare aggiornamenti sui voli prenotati accedendo alla propria area personale, dove possono controllare eventuali ritardi, cancellazioni o variazioni direttamente dall'interfaccia. Inoltre, all'interno della homepage degli utenti viene mostrata una tabella con gli orari aggiornati dei voli in partenza e in arrivo, fornendo una panoramica immediata delle operazioni aeroportuali.

Infine, il sistema permette di eseguire ricerche rapide per trovare voli, passeggeri e bagagli in base a diversi criteri. Le informazioni più importanti vengono evidenziate, come i voli in ritardo o cancellati, per facilitare la gestione delle operazioni Aeroportuali.

Disegno UML:



Spiegazioni delle classe

1. Utente:

La classe utente viene utilizzata per rappresenta tutti gli accessi dei clienti che possono gestire vecchie e nuove prenotazioni. Dispone di due metodi per l'autenticazione dell'utente (login e registrazione) tramite gli attributi **Nome_utente e password**.

Metodi:

- **PrenotaVolo();**
- **VisualizzaVolo();**
- **ModificaVolo();**

- **CercaPasseggero();**
- **Login();**
- **registrazione();**

2. Amministratore:

Questa classe viene utilizzata per distinguere il ruolo dell'amministratore rispetto all'utente normale. Come per l'utente, anche questa classe dispone di un sistema di autenticazione.

Metodi:

- **inserisciVolo();**
- **aggiornaVolo();**
- **Login();**
- **Registrazione().**

3. Volo:

Contiene tutte le informazioni sui voli.

Relazioni:

- **Prenotazione;**
- **StatoVolo;**
- **VoloPartenza;**
- **VoloArrivo;**

4. Prenotazione:

Contiene tutte le prenotazioni effettuate dall'utente.

Relazione:

- **Utente;**
- **Stato prenotazione(enumeration):** Scelta di crearla come enumeration in modo da rendere più automatico il processo di selezione.

5. Documento:

Permette identificare le informazioni del documento del passeggero per la prenotazione.

È un'entità collegata alla classe passeggero.

Attributi:

- **Tipo:** serve a identificare il tipo di documento (carta identità o passaporto), è di tipo **boolean;**
- **Data_scadenza:** data per la validità del documento. È di tipo **Date;**

- **Numero_documento:** numero identificativo del documento. È di tipo **String**;
- **Data_emissione:** data di rilascio del documento. È di tipo **Date**;

6. **Passeggero:**

Permette di gestire le prenotazioni per persone diverse dall'utente che ha effettuato la prenotazione (esempio: familiari).

Relazioni:

Documento: serve per distinguere tra carta d'identità e passaporto, il passaporto con i visti permette di viaggiare in voli internazionali, chi dispone solo della carta d'identità può viaggiare solo nella comunità europea.

7. **voloPartenza e voloArrivo:**

Separano logicamente i voli che partono da quelli che arrivano a Napoli.