

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

DOCUMENTAZIONE PROGETTO OBJECT ORIENTED - BASI DI DATI

Progetto Aeroporto

Vincenzo Raia N86002868, Josuè Emanuel Pugliese N86005374

Anno Accademico: 2024/25

1 Introduzione

1.1 Traccia del progetto e Analisi dei requisiti

1.2 Analisi delle entità e delle relazioni

2 Diagramma delle classi

3 Diagramma di dettaglio delle classi

4 Sequence Diagram

4.1 login(String username, String password)

4.2 creaNuovoVolo(...)

5 GUI (Interfaccia Grafica)

5.1 Ospite

5.2 Utente

5.3 Prenotazione

5.4 SceltaPostoInAereo

5.5 AreaPersonale

5.6 ModificaPrenotazione

5.7 Amministratore

5.8 InserisciVolo

5.9 DettagliVolo

5.10 ModificaVolo

5.11 ModificaGate

1 Introduzione

1.1 Traccia del progetto e Analisi dei requisiti

Il progetto richiede lo sviluppo di 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 consente di organizzare e monitorare le operazioni aeroportuali in modo efficiente e intuitivo.

L'accesso alla piattaforma avviene nella prima schermata, dove gli utenti possono vedere alcune informazioni generiche sui voli disponibili. Dopo aver inserito i dati (email e password) il sistema verifica che le credenziali appartengano ad un profilo esistente per poi visualizzare la successiva pagina in base al ruolo che il profilo ricopre (utente o amministratore).

Gli utenti possono prenotare voli ed entrare nella loro pagina personale, dove potranno visualizzare le prenotazioni effettuate, potendo scegliere di modificarle o cancellarle.

Gli amministratori possono inserire nuovi voli nel sistema, modificare o cancellare voli esistenti oltre a visualizzare informazioni sugli utenti che hanno effettuato prenotazioni per un determinato volo.

1.2 Analisi delle entità e delle relazioni

Entità

Le principali entità che compongono il dominio del problema sono:

- **Ospite**

- id
- email
- password

- **Utente/Amministratore** Rappresenta gli attori che interagiscono con il sistema. Ereditano tutti i loro attributi da Ospite

- **Prenotazione:**

- idPrenotazione
- nome
- cognome
- cartadentita
- email_utente (FK verso Utente)

- **Volo:**

- codiceVolo
- compagniaAerea
- origine

- destinazione
- data
- orarioPrevisto
- ritardo
- email_utente(FK verso Amministratore)
- gate(FK verso Gate)

• Bagaglio:

- codiceBagaglio
- peso
- id_prenotazione(FK verso prenotazione)

• Gate

- numeroGate
- codice_volo(FK verso volo)

• StatoPrenotazione:

- CONFERMATA
- IN_ATTESA
- CANCELLATA

• StatoVolo:

- PROGRAMMATO
- DECOLLATO
- IN_RITARDO
- ATTERRATO
- CANCELLATO

Relazioni

• Effettua (Utente - Prenotazione): 1 a N

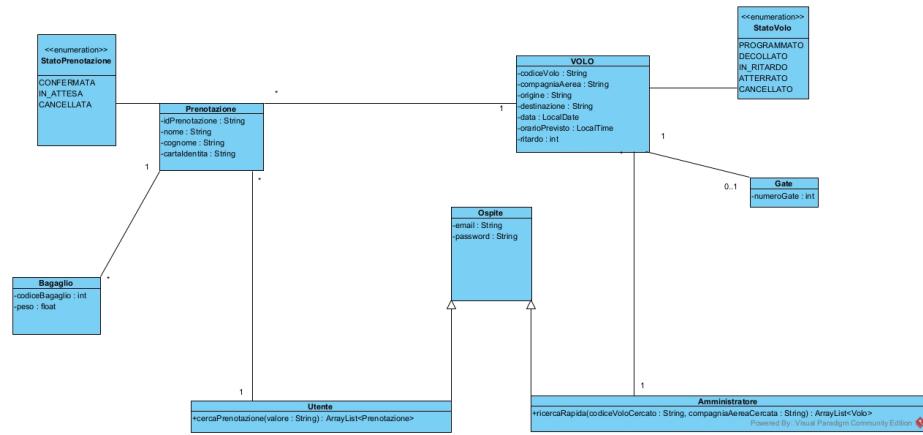
- L'utente effettua più prenotazioni.
- Ogni prenotazione è legata ad un utente.

• Crea (Amministratore - Volo): 1 a N

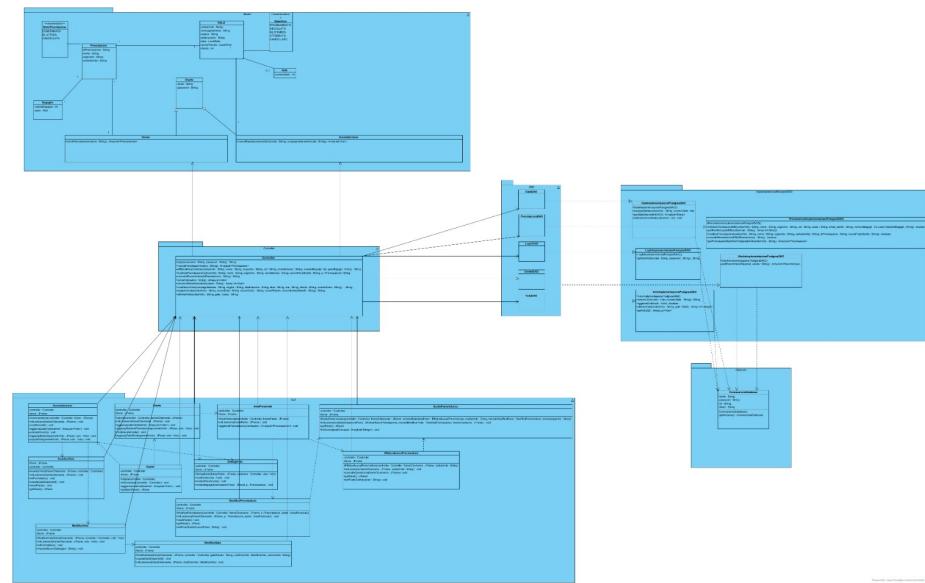
- Un amministratore può creare più voli
- un volo ha associato un solo creatore

- **Assegnato (Volo - Gate):** 1 a 0..1
 - Un volo può essere assegnato ad un solo gate
 - Un gate può avere 0 o 1 volo assegnato
- **Possiede (Prenotazione - Bagaglio):** 1 a N
 - Una prenotazione può contenere più bagagli
 - Un bagaglio ha una sola prenotazione assegnata
- **Associa (Prenotazione - Volo):** N a 1
 - Tabella di associazione:
 - id_prenotazione**(PK, FK verso prenotazione)
 - codice_volo**(FK verso Volo)
 - posto** (il posto scelto dall'utente in aereo)
 - Una prenotazione è associata ad un unico volo
 - Un volo ha più prenotazioni associate ad esso

2 Diagramma delle classi



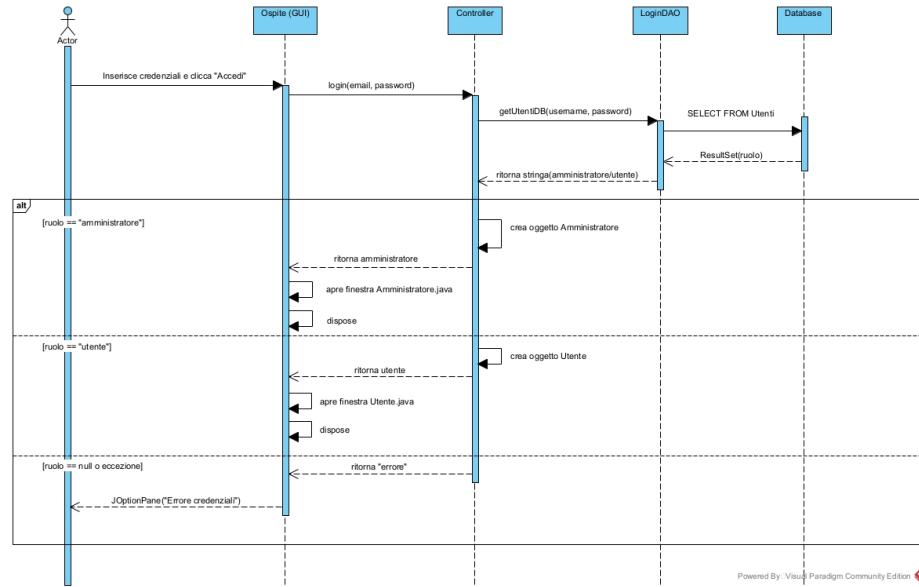
3 Diagramma di dettaglio delle classi



4 Sequence Diagram

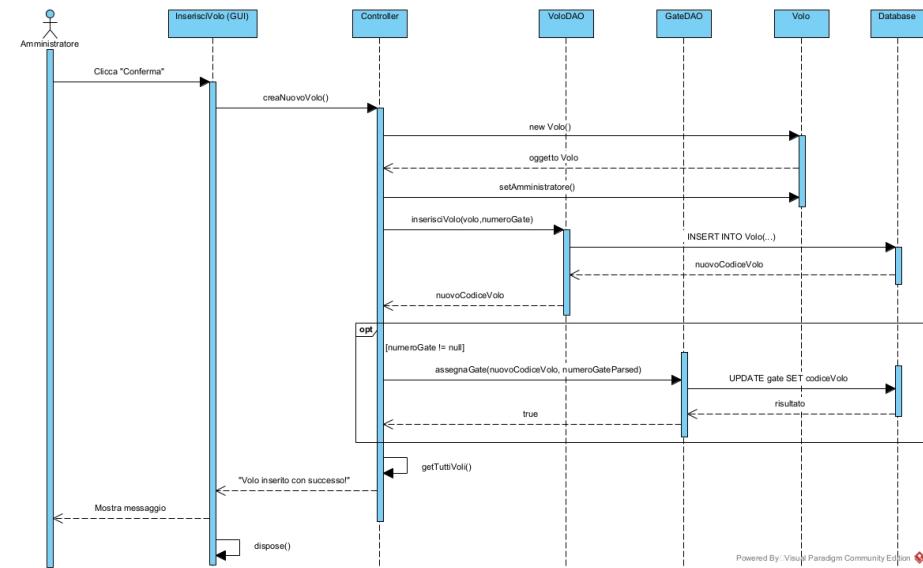
4.1 login(String username, String password)

Il metodo `login(String username, String password)` si occupa di effettuare il corretto login dell'utente verificando se i dati inseriti sono presenti all'interno del database. Il metodo dalla gui chiama il controller (metodo `login`) che a sua volta chiama il database per la verifica dei dati. Quando il database ottiene risultato positivo ritorna il ruolo dell'utente, che viene poi utilizzato dal controller per creare un nuovo oggetto Utente/Amministratore. E' compito della GUI aprire la finestra dell'utente/amministratore sulla base del ruolo che ricopre. Se le credenziali sono sbagliate viene mostrato un messaggio di accesso errato, non permettendo di aprire nessuna nuova pagina.



4.2 inserisciVolo(...)

Il metodo inserisciVolo(String compagnia, String origine, String destinazione, String data, String orario, String ritardo, String gate) permette l'aggiunta di un nuovo volo nel database dell'applicazione. L'amministratore inserisce i dati, che vengono inviati al controller. Il controller crea un nuovo oggetto Volo con i dati inseriti dall'Amministratore. Dopodichè viene inserito all'interno del database chiamando inserisciVolo(...). Viene controllato se è stato inserito un gate da associare al nuovo volo. Se sì, viene aggiornato il database chiamando assegnaGate(...). Un aggiornamento della lista voli viene eseguita dopo le operazioni per aggiornare l'interfaccia grafica.

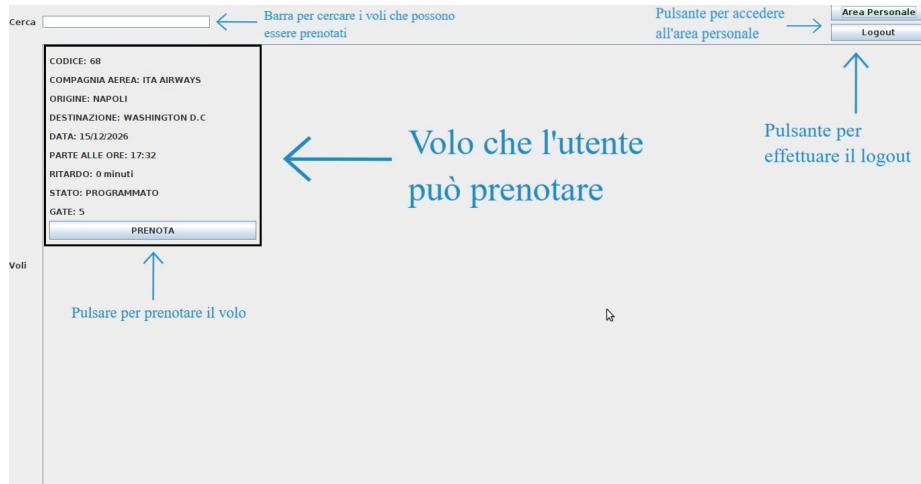


5 GUI

5.1 Ospite



5.2 Utente



5.3 InserisciPrenotazione

Dati prenotazione

INSERISCI PRENOTAZIONE

Nome *

Cognome *

Numero Bagagli *

Carta Identità *

Seleziona il posto in aereo *

Peso totale dei bagagli * KG

Campi per la prenotazione

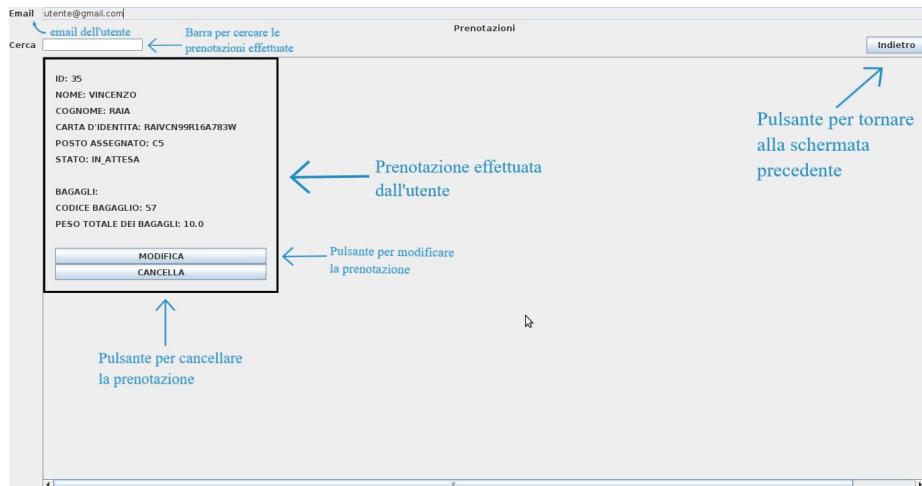
Pulsanti per effettuare o cancellare la prenotazione

12

5.4 SceltaPostoInAereo



5.5 AreaPersonale



5.6 ModificaPrenotazione

Modifica Prenotazione — X

MODIFICA PRENOTAZIONE

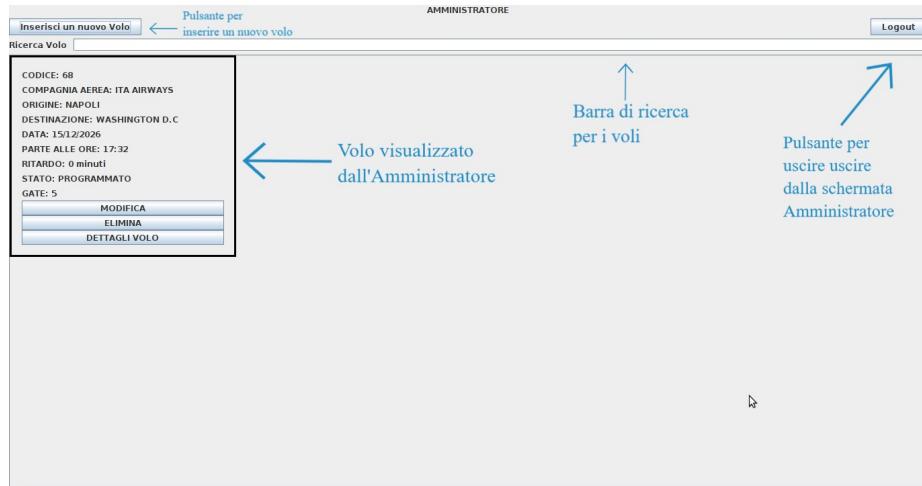
Nome: Campi per ↗

Cognome la prenotazione

Numero Carta Identità
Pulsanti per
posto in aereo modificare 00
una prenotazione ↗

CANCELLA **CONFERMA** **Modifica Posto**

5.7 Amministratore



5.8 InserisciVolo

Schermata InserisciVolo

Compagnia Aerea *	<input type="text"/> Campi di inserimento di un volo
Origine *	<input type="text"/>
Destinazione *	<input type="text"/>
Data del Volo *	<input type="text"/> / /
Orario previsto *	<input type="text"/> :
Ritardo del volo	<input type="text"/> minuti
Gate	<input type="text"/> ▾
Annulla	Conferma
Pulsanti per inserire un volo	Reset

5.9 DettagliVolo

UTENTE:

Nome: Vincenzo

Cognome: Rala

Carta d'identità: ralvcn99r16a783w

BAGAGLI:

Codice Bagaglio: 57

Peso totale dei bagagli: 10.0



Prenotazione associata ad
un volo con i dettagli



5.10 AggiornaVolo



5.11 ModificaGate

