



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
 - cartaIdentita
 - 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

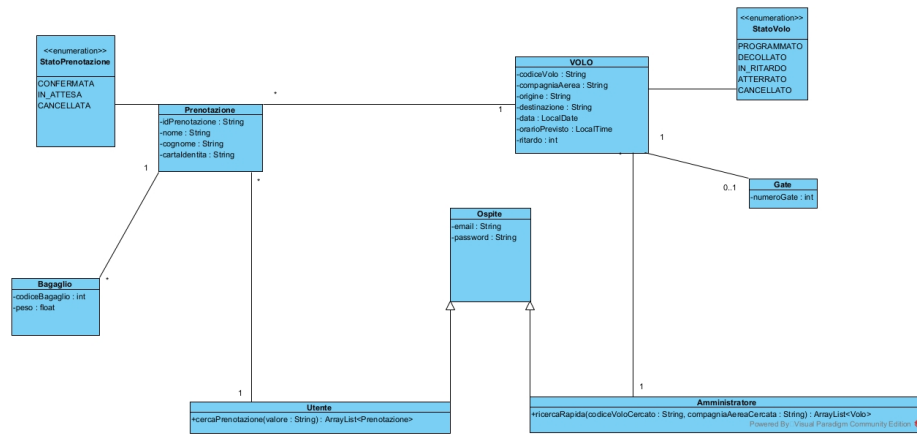
- **Possiede (Prenotazione - Bagaglio):** 1 a N
 - Una prenotazione può contenere più bagagli
 - Un bagaglio ha una sola prenotazione assegnata

- **Associa (Prenotazione - Volo):** N a 1
 - Tabella di associazione:

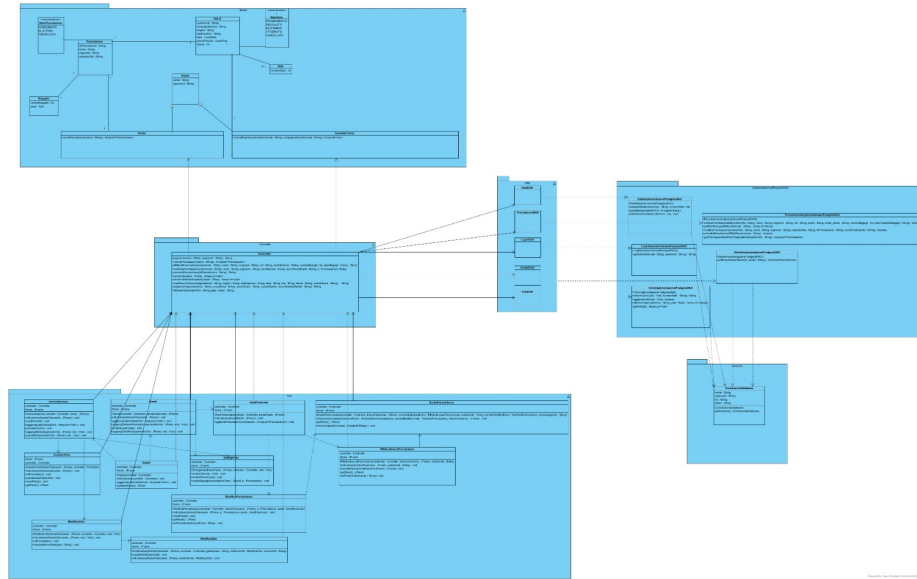
<u>id_prenotazione</u>	(PK, FK verso prenotazione)
<u>codice_volo</u>	(FK verso Volo)
posto	(il posto scelto dall'utente in aereo)

 - Una prenotazione è associata ad un unico volo
 - Un volo ha più prenotazioni associate ad esso

2 Diagramma delle classi



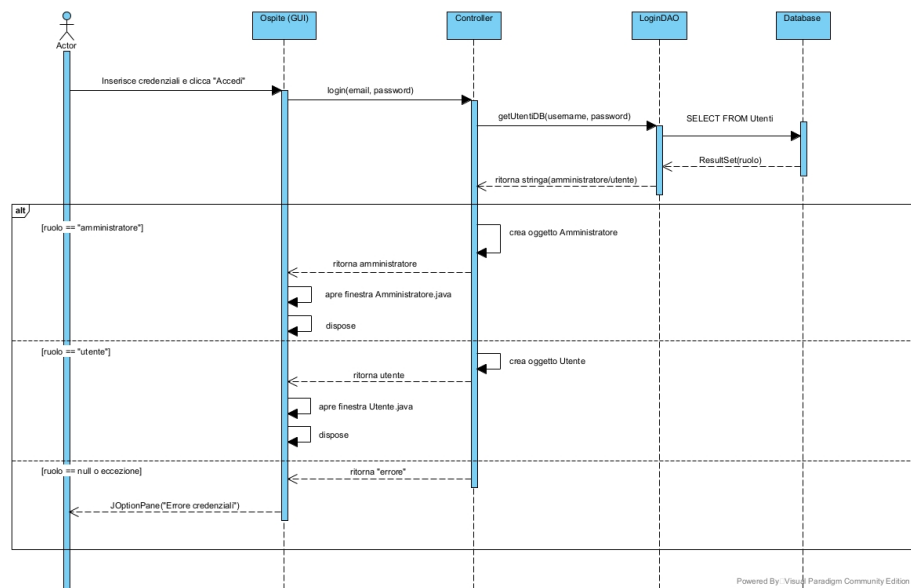
3 Diagramma di dettaglio delle classi



4 Sequence Diagram

4.1 login(String username, String password)

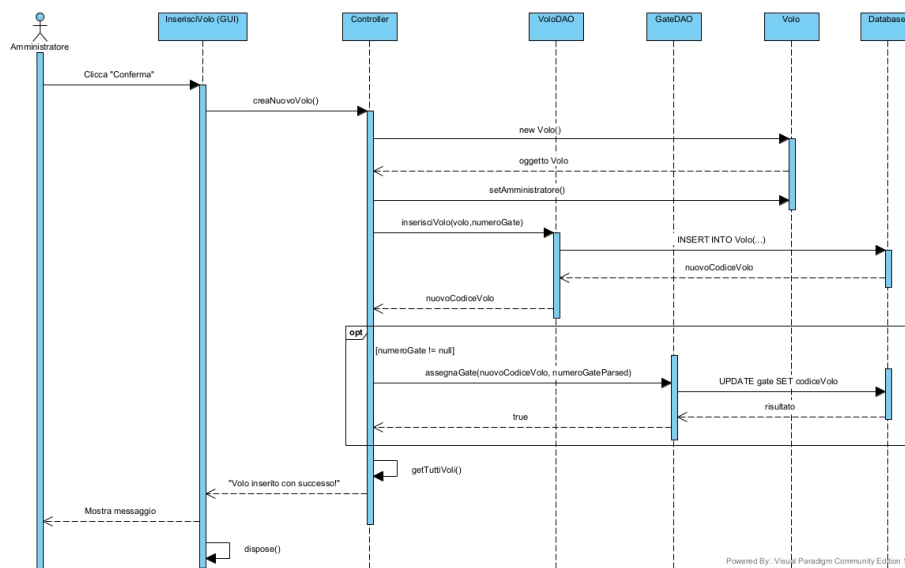
Il metodo login(String username, String password) si occupa di effettuare il corretto login dell'utente verificando se i dati inseriti sono presenti all'interno del database. Il metodo dalla gui chiama il controller (metodo login) che a sua volta chiama il database per la verifica dei dati. Quando il 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.



Powered By: Visual Paradigm Community Edition

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

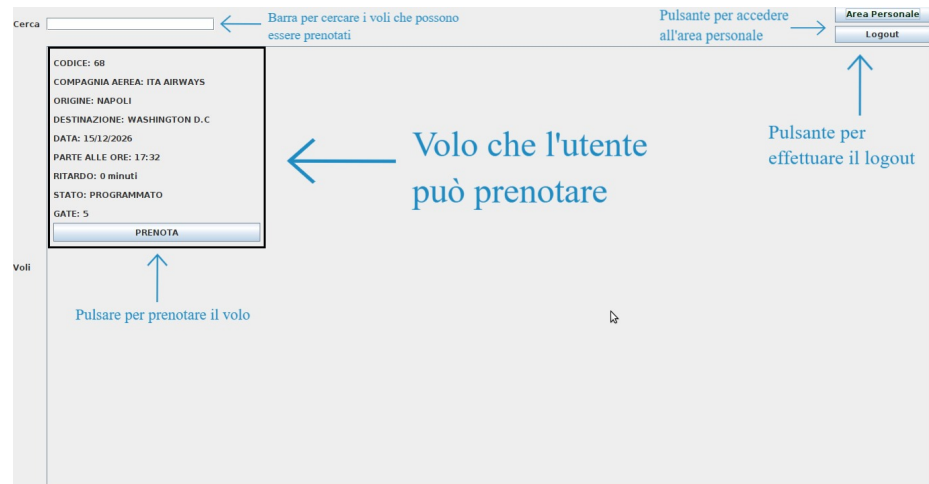
5.1 Ospite

The screenshot shows a Java Swing window titled "La mia GUI Swing". It contains a login section with two text fields labeled "Email" and "Password", and a button labeled "Accedi". A blue arrow points to the "Accedi" button with the text "Pulsante 'Accedi' per effettuare il login". Another blue arrow points to the "Email" and "Password" fields with the text "Campi 'Username' e 'Password'". Below the login section is a table titled "VOLI DISPONIBILI" with the following data:

CODICE	COMPAGNIA AEREA	ORIGINE	DESTINAZIONE	DATA	PARTE ALLE ORE	RITARDO	STATO
0	AIR FRANCE	NAPOLI	TOKYO	12/02/2015	13:23	10 minuti	PROGRAM
7	EASYJET	LONDRA	NAPOLI	11/12/2019	10:30	4 minuti	PROGRAM
5	AIR FRANCE	NAPOLI	TORINO	10/10/2017	11:30	5 minuti	CANCELLA

Below the table is a large grey rectangular area. A blue arrow points to this area with the text "Informazioni del volo".

5.2 Utente



5.3 InserisciPrenotazione

Dati prenotazione

INSERISCI PRENOTAZIONE

Nome *

Cognome *

Numero Bagagli *

Carta Identita *

Peso totale dei bagagli * KG

Seleziona il posto in aereo *

Campi per la prenotazione

Pulsanti per effettuare o cancellare la prenotazione

5.4 SceltaPostoInAereo



5.5 Area Personale

Email: [utente@gmail.com](#)

Cerca:

Barra per cercare le prenotazioni effettuate

Prenotazioni

Indietro

ID: 35
NOME: VINCENTO
COGNOME: RAIA
CARTA D'IDENTITA: RAIVCN99R16A7B3W
POSTO ASSEGNATO: C5
STATO: IN_ATTESA

BAGAGLI:
CODICE BAGAGLIO: 57
PESO TOTALE DEI BAGAGLI: 10.0

MODIFICA
CANCELLA

Prenotazione effettuata dall'utente

Pulsante per modificare la prenotazione

Pulsante per tornare alla schermata precedente

Pulsante per cancellare la prenotazione

5.6 ModificaPrenotazione

The screenshot shows a dialog box titled "Modifica Prenotazione" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog has a light blue header with the title "MODIFICA PRENOTAZIONE". Below the header, there are four input fields for user information: "Nome:", "Cognome", "Numero Carta Identità", and "posto in aereo". The "posto in aereo" field contains the value "00". To the right of the input fields, there are two buttons: "Modifica Posto" and "CONFERMA". At the bottom left, there is a button labeled "CANCELLA".

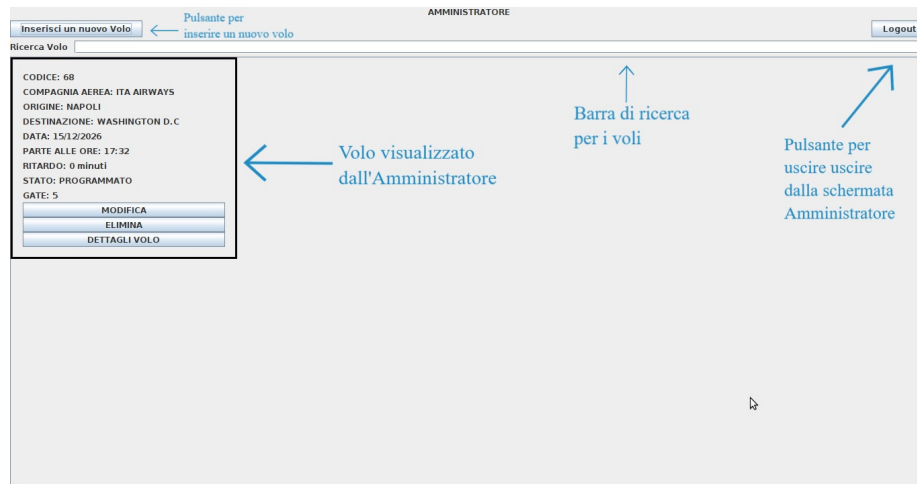
Annotations in blue text with arrows point to the input fields:

- "Campi per la prenotazione" points to the "Nome:" and "Cognome" fields.
- "Pulsanti per modificare una prenotazione" points to the "Modifica Posto" and "CONFERMA" buttons.

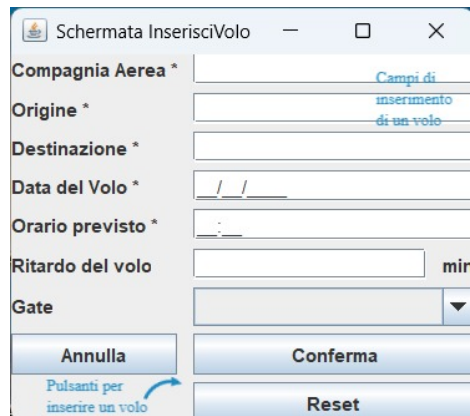
The dialog box contains the following text and elements:

- Title bar: Modifica Prenotazione
- Header: MODIFICA PRENOTAZIONE
- Input fields: Nome:, Cognome, Numero Carta Identità, posto in aereo (00)
- Buttons: Modifica Posto, CANCELLA, CONFERMA

5.7 Amministratore



5.8 InserisciVolo




The screenshot shows a window titled "Schermata InserisciVolo" with standard window controls (minimize, maximize, close). The form contains the following fields and buttons:

- Compagnia Aerea ***: Text input field with a blue link "Campi di inserimento di un volo" to its right.
- Origine ***: Text input field with a blue link "Campi di inserimento di un volo" to its right.
- Destinazione ***: Text input field.
- Data del Volo ***: Text input field with a date mask (/ /).
- Orario previsto ***: Text input field with a time mask (:).
- Ritardo del volo**: Text input field with a "min" label to its right.
- Gate**: Text input field with a dropdown arrow to its right.
- Buttons**: Three buttons at the bottom: "Annulla", "Conferma", and "Reset".

A blue arrow points from the text "Pulsanti per inserire un volo" to the "Conferma" button.

5.9 DettagliVolo

UTENTE: Nome: Vincenzo Cognome: Raia Carta d'identità: raivcn99r16a783w
BAGAGLI: Codice Bagaglio: 57 Peso totale dei bagagli: 10.0


[Prenotazione associata ad
un volo con i dettagli](#)

5.10 AggiornaVolo

Aggiorna Volo

Codice Volo 0

Data Precedente 12/02/2015

Nuova data: / /

Orario del volo: 13:23

Nuovo orario: :

Ritardo: 10

Nuovo Ritardo:

Gate 2

Modifica Gate

Annulla **Conferma**

Campi per l'aggiornamento del volo

pulsate per modificare il gate

Pulsanti per l'aggiornamento dei voli

5.11 ModificaGate

The screenshot shows a window titled "Modifica Gate" with a light gray background. It contains two text input fields and two buttons. The first field is labeled "Gate Attuale" and the second is labeled "Nuovo Gate". Below the fields are two buttons: "Annulla" on the left and "Conferma" on the right. Three blue arrows with text labels point to the fields and the "Conferma" button:

- An arrow points to the "Gate Attuale" field with the label "Campo che indica il gate da modificare".
- An arrow points to the "Nuovo Gate" field with the label "Campo che indica il nuovo gate associato al volo".
- An arrow points to the "Conferma" button with the label "Pulsante per la conferma della modifica del gate".