



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Documentazione Progetto Object Orientation

Myriam Sorrentino N86005248

Francesco Tibello N86005249

Anno accademico 24/25

Indice

1 Traccia	2
2 Analisi dei requisiti	3
2.1 Utente	3
2.2 Amministratore	3
2.3 Generico	4
2.4 Volo	4
2.5 Gate	4
2.6 Bagaglio	4
2.7 Passeggero	5
2.8 Prenotazione	5
3. Diagramma UML	6

1. Traccia: 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.

2 Analisi dei requisiti

In riferimento alla creazione di un'applicazione software in linguaggio Java destinata a supportare le operazioni dell'aeroporto internazionale di Napoli, sono stati definiti i seguenti requisiti funzionali, attraverso l'individuazione di classi specifiche, in linea con le necessità emerse dal contesto:

2.1 Utente

- **Attributi:** login, password
 - **Metodi:** Nessuno direttamente visibile
 - **Descrizione:**
Classe astratta che rappresenta un generico utente del sistema. Viene estesa da due sottoclassi: Utente_generico e Amministratore_del_sistema.
-

2.2 Utente_generico

- **Attributi:** nome
 - **Metodi:**
 - `effettua_pren()` – consente la prenotazione di un volo
 - `visualizza_agg()` – consente la visualizzazione di aggiornamenti
 - `operation()` – operazione generica (da specificare)
 - **Descrizione:**
Rappresenta l'utente standard del sistema, ovvero colui che può effettuare prenotazioni e monitorare i propri voli.
Esempio: Mario Rossi accede al sistema per prenotare un volo per Milano e controlla lo stato del volo prenotato.
-

2.3 Amministratore_del_sistema

- **Attributi:** nessuno aggiuntivo, eredita da Utente
 - **Metodi:**
 - `inserisci_volo()` – consente di inserire un nuovo volo
 - `update_volo()` – consente di aggiornare le informazioni di un volo
 - `mod_gate()` – permette la modifica dell'assegnazione del gate
 - **Descrizione:**

Rappresenta il personale autorizzato alla gestione e aggiornamento dei dati relativi a voli e gate.

Esempio: Un amministratore modifica il gate del volo AZ123 da 12 a 15.
-

2.4 Volo

- **Attributi:** codice, compagnia aerea, data, orario previsto, ritardo, origine, destinazione, stato
 - **Metodi:** Nessuno visibile
 - **Descrizione:**

Contiene tutte le informazioni relative a un volo, incluse compagnia, orario e stato. Lo stato è rappresentato da un enum `stato_del_volo` con valori: `inviato`, `in_ritardo`, `in_orario`, `cancellato`, `atterrato`.

Esempio: Il volo LH567 è previsto per le 14:00, ma ha un ritardo di 45 minuti e risulta "in_ritardo".
-

2.5 Volo_partenza

- **Attributi:** origine = Napoli, gate_partenza (riferimento alla classe Gate)
 - **Descrizione:**

Specializzazione della classe Volo che rappresenta un volo in partenza da Napoli.
-

2.6 Volo_arrivo

- **Attributi:** destinazione = Napoli
 - **Descrizione:**

Specializzazione della classe Volo che rappresenta un volo in arrivo a Napoli.
-

2.7 Gate

- **Attributi:** numero
 - **Descrizione:**
Rappresenta un gate d'imbarco, identificato da un numero intero. Può essere assegnato a uno o più voli in partenza.
Esempio: Il gate numero 8 è assegnato al volo Alitalia AZ321.
-

2.8 Prenotazione

- **Attributi:** PNR, num_biglietto, stato_pren (enum: confermata, in_attesa, cancellata), assicurazione (boolean), bagaglio (boolean)
 - **Descrizione:**
Rappresenta una prenotazione fatta da un utente per un volo. Include informazioni sul biglietto e servizi aggiuntivi (assicurazione e bagagli).
Esempio: Prenotazione con PNR AZ4567, stato "confermata", con bagaglio e assicurazione inclusi.
-

2.9 Passeggero

- **Attributi:** nome, cognome, posto, email, num_telefono (String), SSN (numero identificativo)
 - **Descrizione:**
Rappresenta la persona fisica che viaggia. I dati possono essere diversi da quelli dell'utente che ha effettuato la prenotazione.
Esempio: L'utente Maria Bianchi prenota un volo per suo padre, Giuseppe Bianchi, che viaggia nel posto 12B.
-

3. Diagramma UML

Di seguito viene presentato il diagramma UML, che utilizza un formalismo standard per modellare la struttura e le relazioni del dominio del problema, fornendo una rappresentazione visiva delle classi e delle loro interconnessioni.

