



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Documentazione Progetto Object Orientation

Carmine Sgariglia, N86005069

Mattia Lemma, N86005170

Massimo Russo, N86005016

Anno accademico 24/25

Indice

1	Traccia	2
2	Analisi dei requisiti	3
2.0.1	Utente	3
2.0.2	Amministratore	3
2.0.3	Generico	3
2.0.4	Volo	4
2.0.5	Gate	4
2.0.6	Bagaglio	4
2.0.7	Passeggero	5
2.0.8	Prenotazione	5
2.1	Diagramma UML	6

1 Traccia

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. Un'altra funzione importante è il monitoraggio dei bagagli. Ogni bagaglio è associato al volo del passeggero e viene registrato nel sistema durante l'operazione di check-in, generando un codice univoco che consente il tracciamento. Durante il percorso, lo stato del bagaglio viene gestito manualmente dagli amministratori, che aggiornano il sistema indicando

se è stato caricato sull'aereo o è disponibile per il ritiro. Gli utenti possono visualizzare lo stato aggiornato del proprio bagaglio tramite l'interfaccia del sistema. Se un bagaglio viene smarrito, l'utente può segnalarlo direttamente nel sistema. Gli amministratori possono accedere a un modulo dedicato per visionare le richieste di smarrimento.

2 Analisi dei requisiti

Per la realizzazione di un applicativo Java utile all'aeroporto internazionale di Napoli in accordo ai requirements sono state individuate le seguenti classi:

2.0.1 Utente

- **Attributi:** nomeUtente, password, nome, cognome.
- **Metodi:** VisualizzaVoli();
- **Descrizione:** Utente generalizza le due classi di utenti(amministratore e generico) presenti nella piattaforma.

2.0.2 Amministratore

- **Attributi:** Eredita quelli della superclasse Utente.
- **Metodi:** Eredita quello della superlcasse Utente (VisualizzaVoli()), inserisciVolo(), modificaVolo(). assegnaGate(), aggiornaBagaglio(), visualizzaSmarriti().
- **Descrizione:** Amministratore è la specializzazione di Utente che può amministrare i contenuti della piattaforma legati a voli e prenotazioni.

2.0.3 Generico

- **Attributi:** Eredita quelli della superclasse Utente.
- **Metodi:** Eredita quello della superlcasse Utente (VisualizzaVoli()), prenota(), cercaPrenotazione(codiceVolo), cercaPrenotazione(nomePasseggero, cognomePasseggero), segnalaSmarrimento().
- **Descrizione:** Generico è la specializzazione di Utente che rappresenta il consumatore che può prenotare e gestire i propri voli

2.0.4 Volo

- **Attributi:** codicevolo, compagnia, origine, destinazione, orarioPrevisto, stato, data, tempoRitardo.
- **Metodi:** Non ci sono particolari metodi individuati nella traccia per questa classe.
- **Descrizione:** Volo ci permette di identificare i voli operati da e per l'aeroporto internazionale di Napoli.
- Lo stato del volo è rappresentato attraverso un **enum** che può assumere i seguenti valori: programmato, inRitardo, decollato, atterrato, cancellato.

2.0.5 Gate

- **Attributi:** codice (deve essere di tipo intero in accordo con i requisiti della traccia fornita dal cliente).
- **Metodi:** Non ci sono particolari metodi individuati nella traccia per questa classe.
- **Descrizione:** Gate identifica il numero del gate dei viaggi in partenza dall'aeroporto internazionale di Napoli.

2.0.6 Bagaglio

- **Attributi:** codice, stato.
- **Metodi:** Non ci sono particolari metodi individuati nella traccia per questa classe.
- **Descrizione:** Bagaglio identifica i bagagli posseduti dai viaggiatori durante i loro viaggi.
- Lo stato del bagaglio è rappresentato attraverso un **enum** che può assumere i seguenti valori: caricato, ritirabile.

2.0.7 Passeggero

- **Attributi:** nome, cognome, nDocumento.
- **Metodi:** Non ci sono particolari metodi individuati nella traccia per questa classe.
- **Descrizione:** Passeggero identifica le effettive persone titolari di una carta d'imbarco(possono differire rispetto ai nomi utenti).

2.0.8 Prenotazione

- **Attributi:** numeroBiglietto, posto, stato.
- **Metodi:** checkin().
- **Descrizione:** Prenotazione identifica tutte le prenotazioni dei voli eseguite dagli utenti.
- Lo stato del bagaglio è rappresentato attraverso un **enum** che può assumere i seguenti valori: confermato, inAttesa, cancellato.

2.1 Diagramma UML

Di seguito forniamo il diagramma con formalismo UML utilizzato per modellare il dominio del problema:

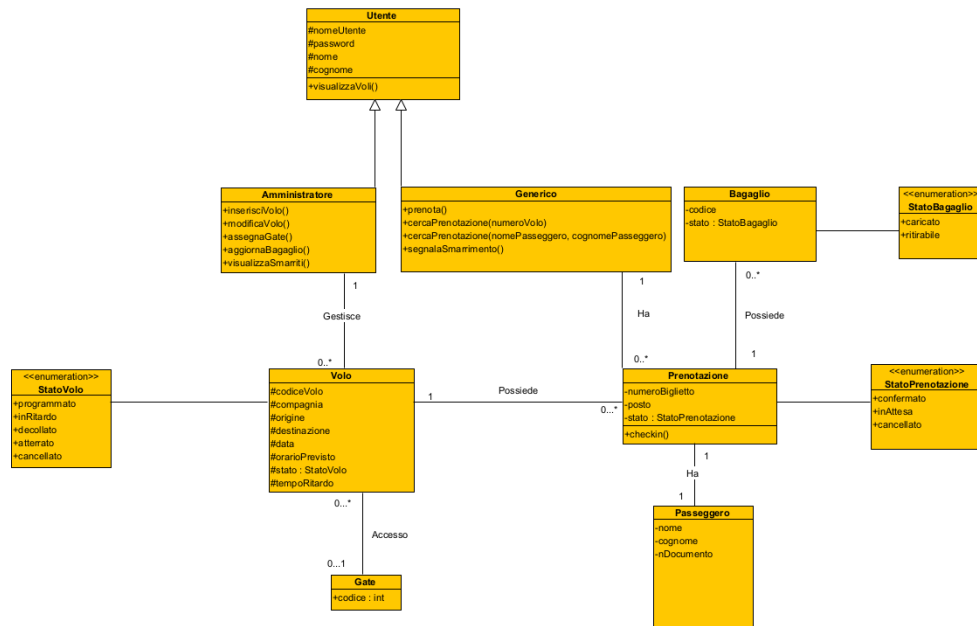


Figura 1: Diagramma UML