



UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Documentazione Progetto Object Orientation

Carmine Sgariglia (N86005069)

Mattia Lemma (N86005170)

Massimo Russo (N86005016)

Anno accademico 2024/2025

Indice

1	Introduzione	3
1.1	Traccia del progetto	3
1.2	Analisi dei requisiti	4
1.2.1	Utente	4
1.2.2	Amministratore	4
1.2.3	Generico	5
1.2.4	Volo	5
1.2.5	Gate	5
1.2.6	Bagaglio	5
1.2.7	Passeggero	6
1.2.8	Prenotazione	6
1.2.9	Relazioni	6
2	Diagramma UML delle classi del dominio del problema	7
3	Diagramma UML di dettaglio delle classi nel dominio della soluzione	8
4	Sequence diagram di funzionalità a scelta	9
4.1	aggiornaBagaglio(bagaglio)	9
4.2	getAllGates()	10
5	Manuale della GUI	12
5.1	Schermate introduttive	12
5.1.1	Interfaccia di scelta login/registrazione	12
5.1.2	Interfaccia di Login	13
5.1.3	Interfaccia di Registrazione	14
5.2	Admin	15
5.2.1	Interfaccia principale Voli	15
5.2.2	Interfaccia dettagli volo	16
5.2.3	Interfaccia di gestione prenotazioni	17
5.2.4	Interfaccia modifica prenotazioni	18
5.2.5	Interfaccia inserimento volo	19

5.2.6	Interfaccia bagagli smarriti	20
5.3	Utente	21
5.3.1	Interfaccia Visualizzazione e ricerca voli	21
5.3.2	Interfaccia prenotazione voli	22
5.3.3	Interfaccia gestione prenotazioni effettuate	23
5.3.4	Interfaccia monitoraggio bagagli	24
6	Gestione errori	25

Link repository :

https://github.com/Massimo127r/Applicativo_Aeroporto.git

Capitolo 1

Introduzione

1.1 Traccia del progetto

1. Obiettivo del Sistema

Il sistema informativo sviluppato ha come obiettivo la gestione centralizzata ed efficiente delle operazioni aeroportuali dell'aeroporto di Napoli. L'architettura del sistema si compone di due elementi principali: una base di dati relazionale, per la memorizzazione strutturata delle informazioni, e un'applicazione desktop realizzata in Java, con interfaccia grafica costruita mediante la libreria Swing.

2. Categorie di Utenti e Accesso

Il sistema è progettato per due categorie distinte di utenti: utenti generici e amministratori di sistema. L'accesso all'applicazione avviene tramite autenticazione con credenziali personali (login e password). Ogni utente dispone di privilegi specifici, in base al proprio ruolo.

3. Funzionalità per gli Utenti Generici

Gli utenti generici possono consultare i voli programmati ed effettuare prenotazioni per sé o per altri passeggeri. Ogni prenotazione include il numero del biglietto, il posto assegnato e lo stato della prenotazione (confermata, in attesa, cancellata). Il sistema consente inoltre la ricerca e modifica delle prenotazioni attraverso filtri basati sul nome del passeggero o sul codice del volo.

4. Funzionalità per gli Amministratori

Gli amministratori, oltre ad avere accesso a tutte le funzionalità degli utenti generici, possono inserire nuovi voli e aggiornare i dati dei voli esistenti. Ogni volo è identificato da un codice univoco e include informazioni quali compagnia aerea, aeroporto di origine (per voli in arrivo) o destinazione (per voli in partenza), data e orario previsto, eventuale ritardo, stato attuale (programmato, decollato, in ritardo, atterrato, cancellato) e gate d'imbarco (per i voli in partenza).

5. Gestione Bagagli

Il sistema prevede una sezione dedicata alla gestione e tracciamento dei bagagli. Ogni bagaglio è associato a una prenotazione, identificato da un codice univoco generato al momento del check-in, e il suo stato viene aggiornato dagli amministratori (ad esempio: registrato, caricato, disponibile al ritiro). Gli utenti possono monitorare in tempo reale la posizione e lo stato del proprio bagaglio. È inoltre possibile segnalare lo smarrimento tramite una funzione dedicata, con gestione riservata agli amministratori.

6. Homepage e Panoramica Operativa

La homepage dell'applicazione mostra una panoramica aggiornata delle operazioni aeroportuali in corso, tramite una tabella che visualizza voli in arrivo e in partenza, con particolare evidenza per ritardi o cancellazioni. Questa funzionalità garantisce una gestione organizzata, tempestiva ed efficiente delle attività aeroportuali, migliorando l'esperienza dell'utente.

7. Strumenti di Ricerca Avanzata

Il sistema include strumenti di ricerca avanzata che permettono di filtrare rapidamente voli, prenotazioni, passeggeri e bagagli. Questa funzionalità risulta utile soprattutto in scenari complessi o ad alto volume di traffico, rendendo l'interazione con il sistema rapida ed efficace.

1.2 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:

1.2.1 Utente

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

1.2.2 Amministratore

- **Attributi:** Eredita quelli della superclasse Utente.
- **Metodi:** Eredita quello della superclasse 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.

1.2.3 Generico

- **Attributi:** Eredita quelli della superclasse Utente.
- **Metodi:** Eredita quello della superclasse 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

1.2.4 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.

1.2.5 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.

1.2.6 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.

1.2.7 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).

1.2.8 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.

1.2.9 Relazioni

- **Gestisce** (Amministratore-Volo)
 - Ogni amministratore può gestire 0 o più voli.
- **Ha** (Generico-Prenotazione)
 - Ogni utente può avere 0 o più prenotazioni.
- **Possiede** (Prenotazione-Bagaglio)
 - Ogni prenotazione può aver ad essa associato 0 o più bagagli.
- **Accesso** (Volo-Gate)
 - Ogni volo ha accesso a 0 o 1 gate.
- **Ha** (Prenotazione-Passeggero)
 - Ogni prenotazione ha un passeggero.

Capitolo 2

Diagramma UML delle classi del dominio del problema

Di seguito forniamo un diagramma, realizzato secondo il formalismo UML, utilizzato per modellare il dominio del problema. Il diagramma delle classi rappresenta i principali concetti coinvolti nel sistema informativo per la gestione dell'aeroporto, evidenziando le entità, gli attributi rilevanti e le relazioni tra gli oggetti che compongono l'architettura logica dell'applicazione.



Figura 2.1: Diagramma UML del dominio del problema

Capitolo 3

Diagramma UML di dettaglio delle classi nel dominio della soluzione

Il seguente diagramma, costruito secondo il formalismo UML, rappresenta il dominio della soluzione e descrive l'architettura logica del sistema informativo così come implementato. In particolare, il diagramma evidenzia le principali classi software dell'applicazione, con i relativi attributi e metodi, nonché le relazioni statiche tra di esse (come associazioni, aggregazioni e dipendenze). Vengono inoltre riportati i legami con la struttura dei dati persistenti, offrendo una visione chiara della corrispondenza tra il modello concettuale e la sua realizzazione concreta nel codice. Questo modello è stato progettato seguendo i principi di separazione delle responsabilità e modularità, facilitando così la manutenibilità, l'estensibilità e il riutilizzo del software. In particolare, si è adottata un'architettura a livelli, che distingue logicamente tra interfaccia utente, logica di funzionamento e accesso ai dati.

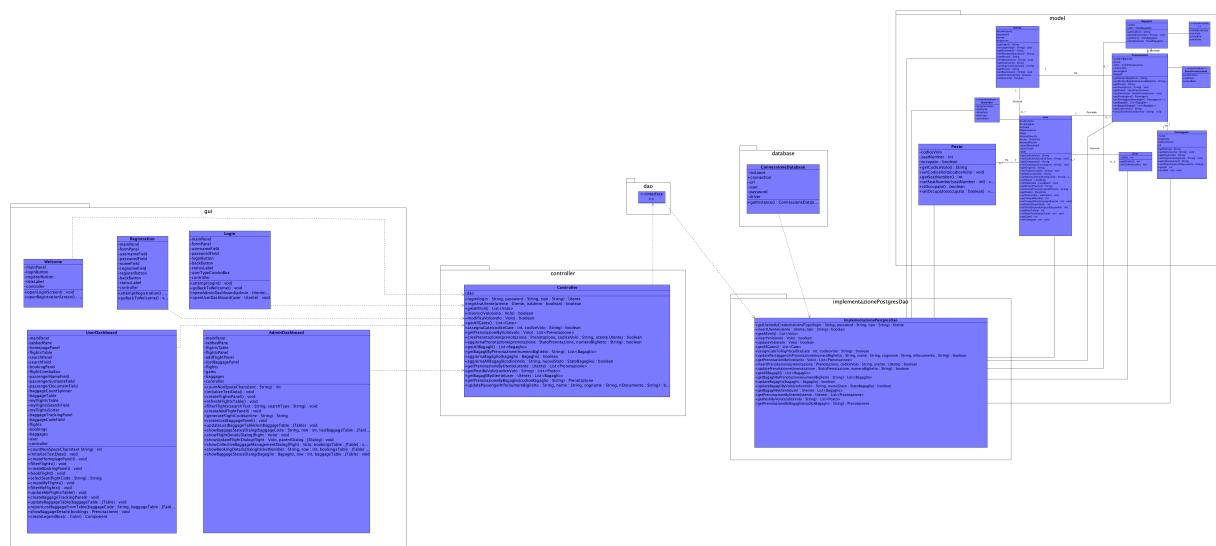


Figura 3.1: Diagramma UML del dominio della soluzione

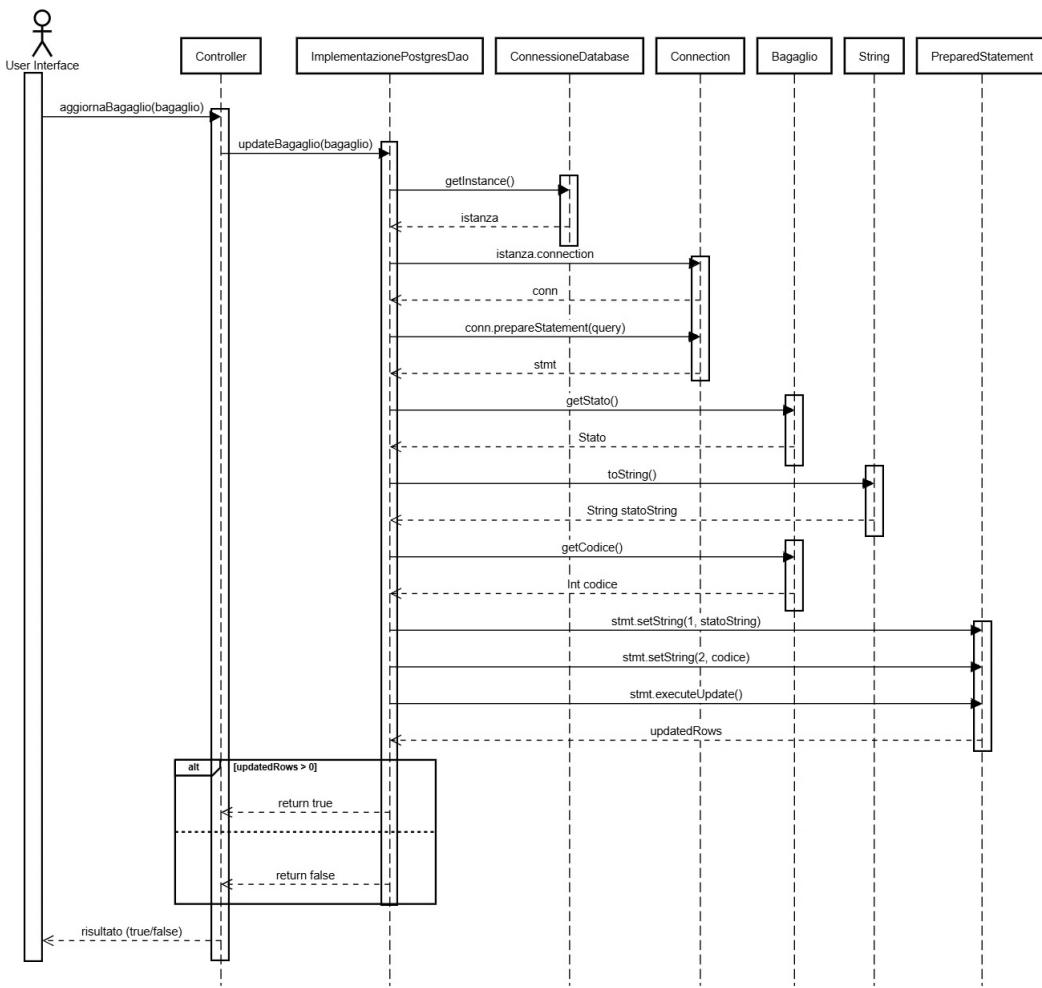
Capitolo 4

Sequence diagram di funzionalità a scelta

Di seguito vengono presentati due diagrammi di sequenza UML, selezionati per illustrare in modo puntuale alcune delle principali funzionalità offerte dall'applicativo. Tali diagrammi descrivono l'interazione dinamica tra gli oggetti del sistema durante l'esecuzione di specifici casi d'uso, evidenziando l'ordine temporale dei messaggi scambiati e la responsabilità dei diversi componenti. L'obiettivo è fornire una rappresentazione chiara del comportamento del sistema in risposta a determinati eventi, mettendo in luce il flusso delle informazioni tra interfaccia utente, logica applicativa e livello di persistenza.

4.1 aggiornaBagaglio(bagaglio)

Il seguente diagramma di sequenza descrive l'interazione tra i componenti del sistema durante l'esecuzione del metodo `aggiornaBagaglio(bagaglio)`. Questa funzionalità, accessibile agli amministratori, consente di modificare lo stato di un bagaglio già registrato, ad esempio per indicare se è stato caricato sull'aereo o reso disponibile per il ritiro. Il diagramma evidenzia il flusso dei messaggi tra l'interfaccia grafica, il controller applicativo e il livello di persistenza, mostrando come l'oggetto `bagaglio` venga passato e aggiornato nel database in modo coerente e tracciabile.


 Figura 4.1: Sequence Diagram `aggiornaBagaglio(bagaglio)`

4.2 getAllGates()

Il diagramma di sequenza che segue rappresenta l'esecuzione del metodo `getAllGates()`, utilizzato per recuperare l'elenco completo dei gate disponibili presso l'aeroporto. Questa operazione può essere invocata, ad esempio, durante la fase di assegnazione di un gate a un volo in partenza. Il diagramma mostra l'interazione tra i principali componenti del sistema: l'interfaccia grafica che attiva la richiesta, il controller che coordina la logica dell'operazione e il livello di accesso ai dati che esegue la query sul database. Il metodo restituisce una collezione di oggetti `Gate`, ciascuno dei quali rappresenta un gate identificato da un numero univoco, rendendoli disponibili per la visualizzazione e la selezione da parte dell'amministratore.

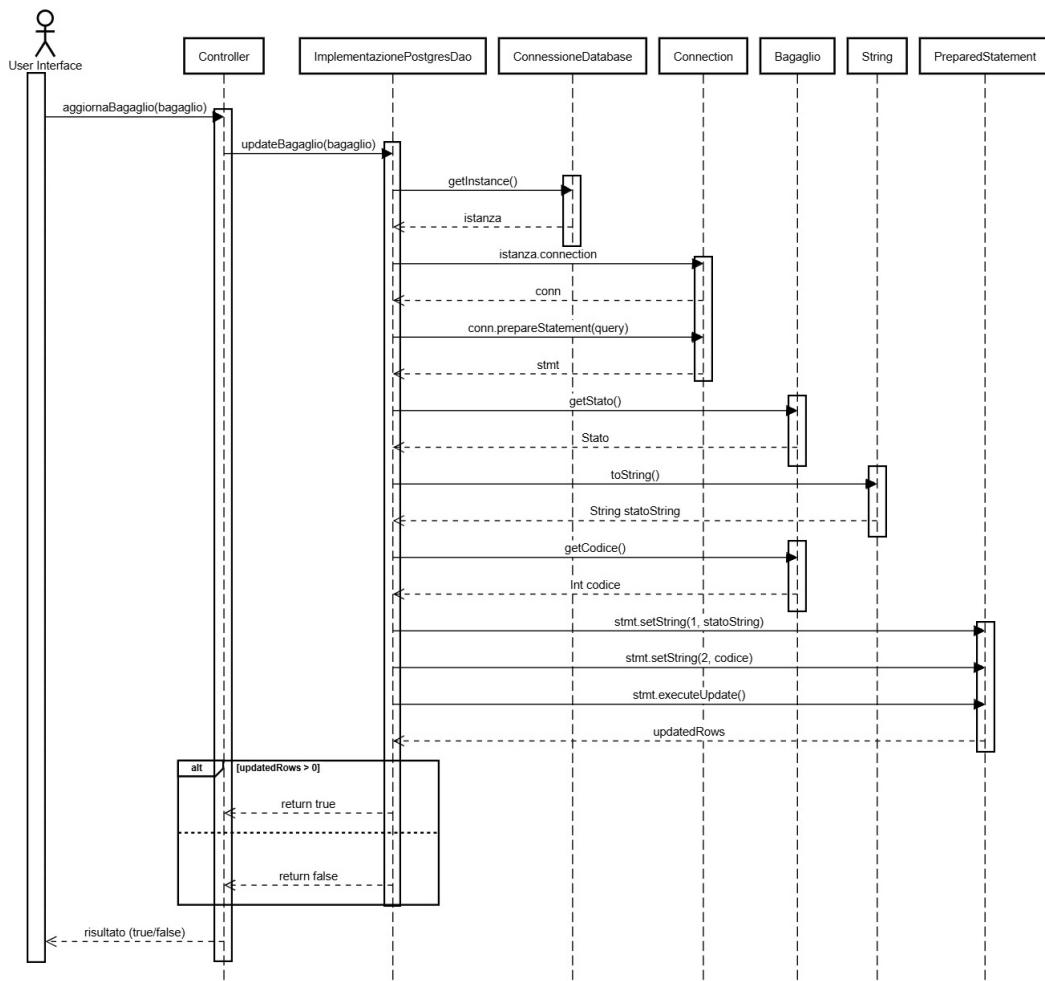


Figura 4.2: Sequence Diagram `addAllGates()`

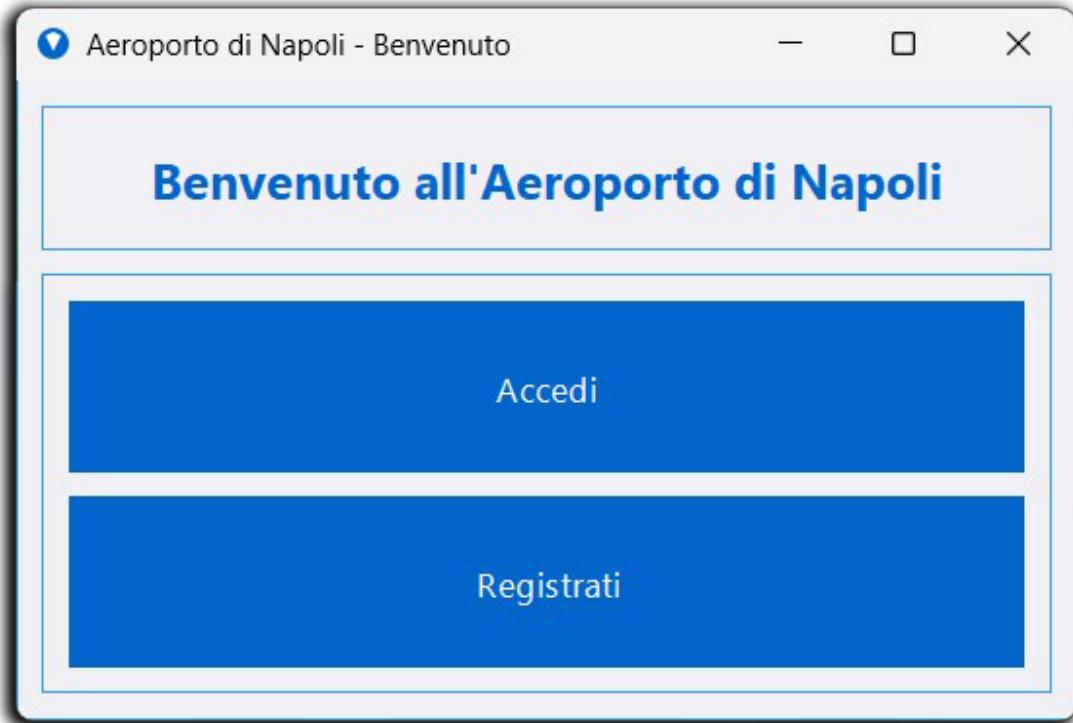
Capitolo 5

Manuale della GUI

5.1 Schermate introduttive

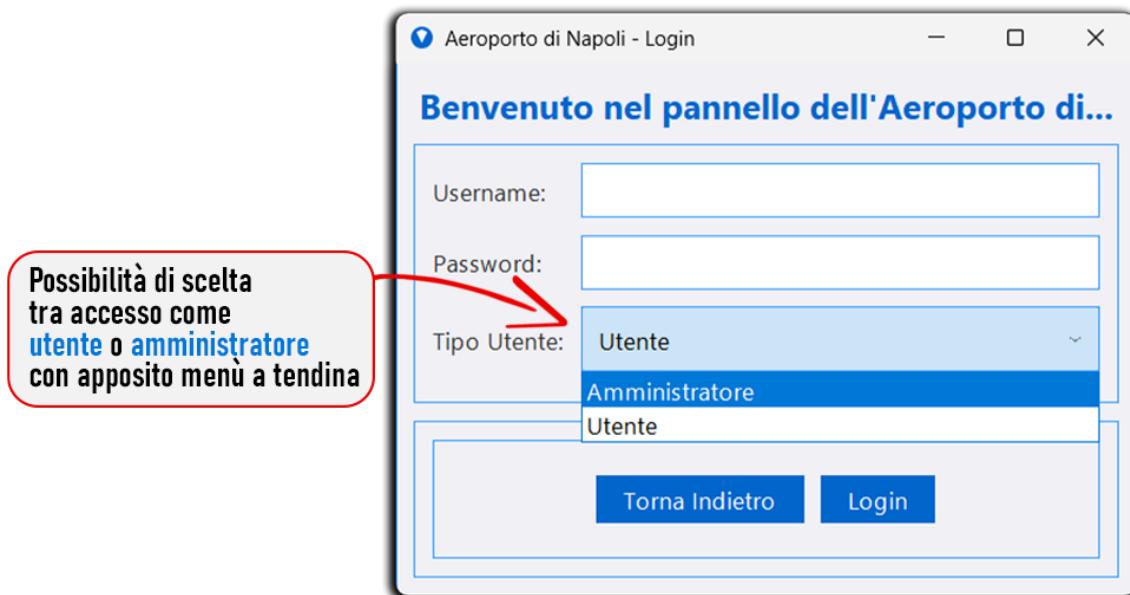
5.1.1 Interfaccia di scelta login/registrazione

All'avvio dell'applicazione viene presentata una schermata di benvenuto, dalla quale l'utente può scegliere se procedere con il login o con la registrazione di un nuovo account. Questa interfaccia funge da punto di ingresso al sistema e garantisce una separazione chiara tra utenti già registrati e nuovi utenti.



5.1.2 Interfaccia di Login

L'interfaccia di login consente l'autenticazione degli utenti tramite l'inserimento delle credenziali (username e password). Un apposito menu a tendina permette di selezionare il tipo di utente (Amministratore o Utente generico), in modo da indirizzare correttamente verso la dashboard corrispondente. Questa distinzione è fondamentale per l'abilitazione delle diverse funzionalità previste per ciascun ruolo.



5.1.3 Interfaccia di Registrazione

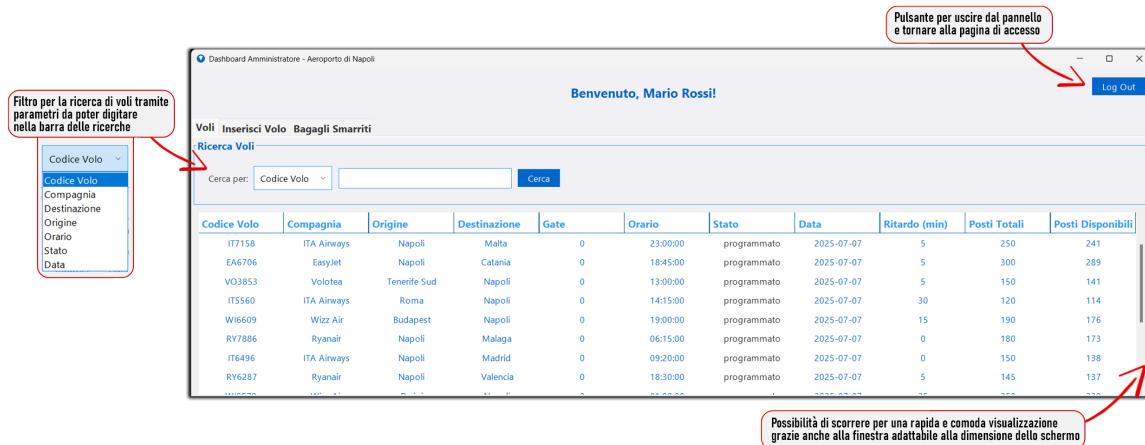
L'interfaccia di registrazione consente la creazione di un nuovo account utente mediante l'inserimento obbligatorio delle seguenti informazioni: username, password, nome e cognome. È previsto un pulsante per tornare alla schermata di login in caso di errore o ripensamento. Il sistema verifica che tutti i campi siano compilati correttamente prima di consentire la registrazione, garantendo la consistenza dei dati utente.



5.2 Admin

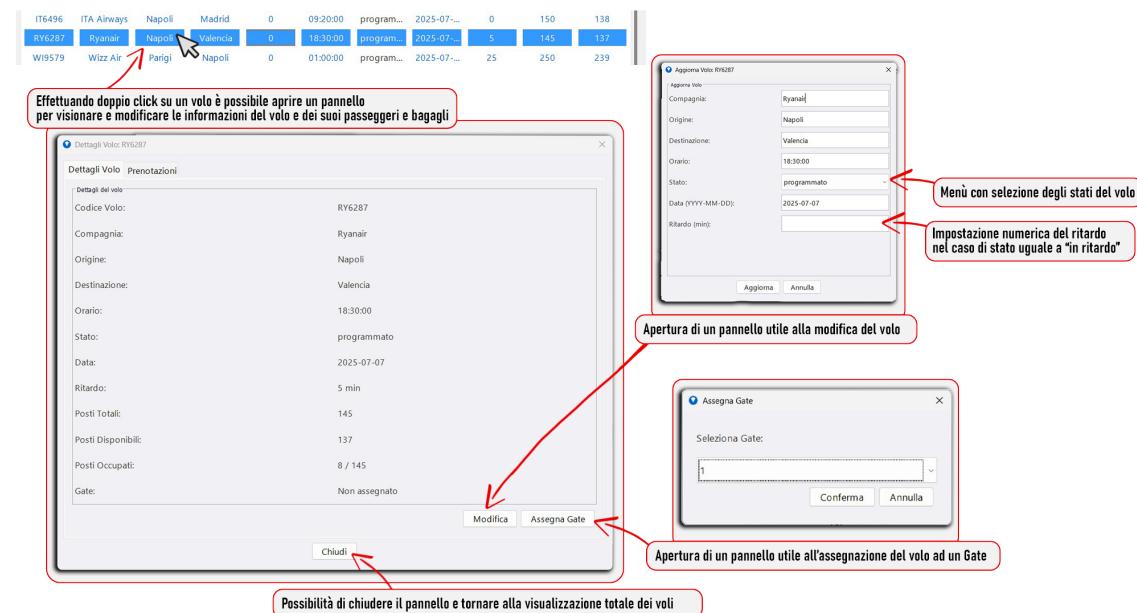
5.2.1 Interfaccia principale Voli

La schermata principale del pannello amministratore mostra l'elenco completo dei voli presenti nel sistema. Tramite il motore di ricerca filtrabile per codice, compagnia, origine, destinazione, orario, stato o data, l'amministratore può individuare rapidamente i voli di interesse. I dati visualizzati comprendono inoltre informazioni operative, come ritardo previsto, numero di posti totali e disponibili. L'interfaccia è progettata per essere adattabile e scorrevole, anche su display di dimensioni ridotte.



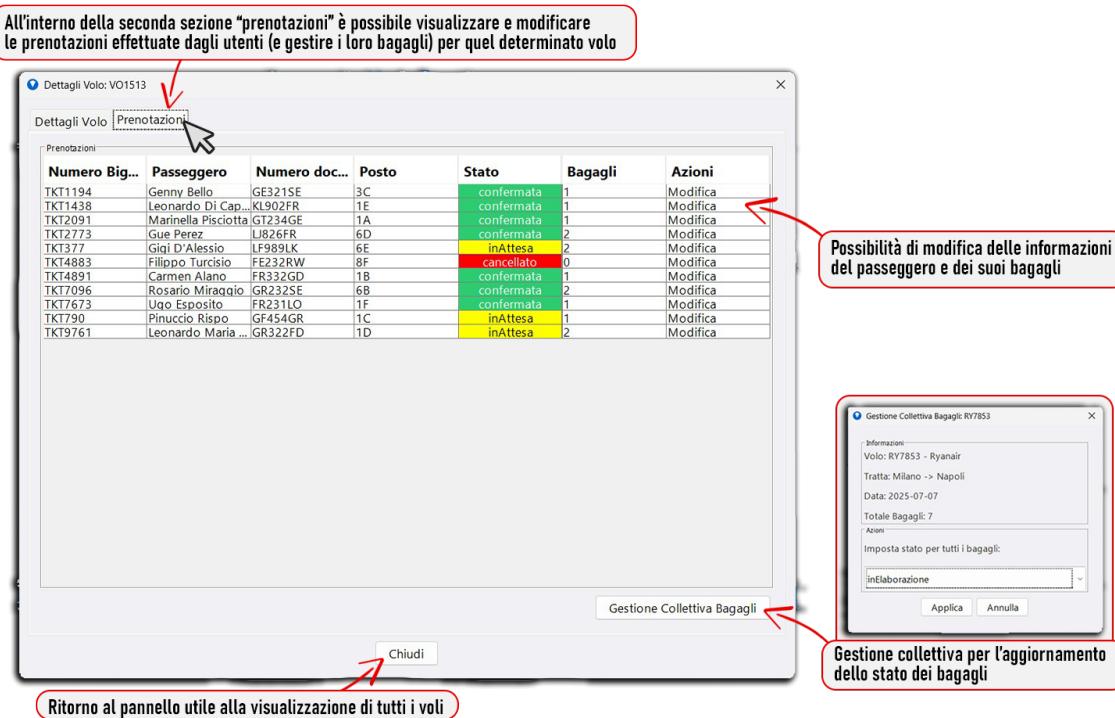
5.2.2 Interfaccia dettaglio volo

A seguito di un doppio clic su una riga dell'elenco voli, viene aperta una finestra di dettaglio del volo selezionato. Da qui è possibile visualizzare tutte le informazioni aggiornate relative al volo, accedere alla sezione delle prenotazioni e dei bagagli, oppure modificare lo stato del volo e assegnare un gate tramite pannelli dedicati. Questa interfaccia rappresenta il punto di accesso centrale per l'amministrazione delle operazioni legate a un volo specifico.



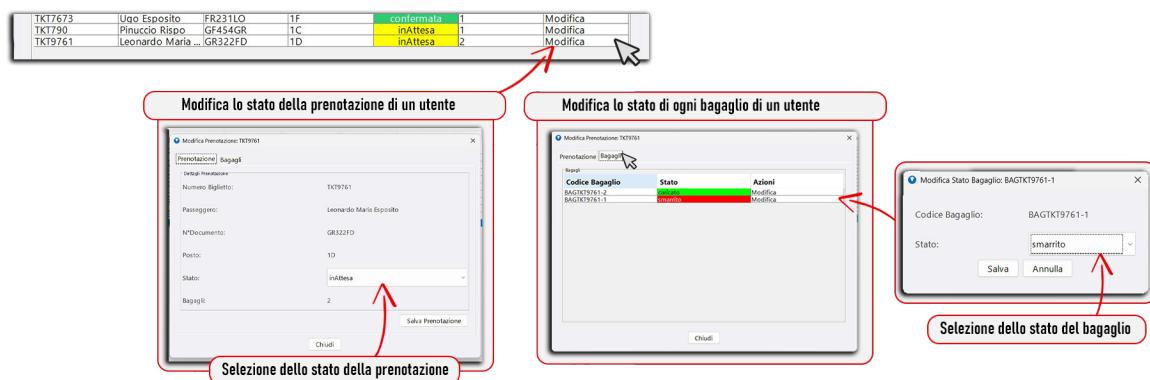
5.2.3 Interfaccia di gestione prenotazioni

Questa interfaccia consente agli amministratori di visualizzare e gestire le prenotazioni relative a un determinato volo. Attraverso la scheda "Prenotazioni", è possibile accedere all'elenco dei passeggeri, visualizzare i dettagli delle prenotazioni (stato, posto, numero bagagli) ed effettuare modifiche puntuali per ciascun passeggero. È inoltre disponibile un'apposita funzione di gestione collettiva per l'aggiornamento simultaneo dello stato di tutti i bagagli associati al volo selezionato.



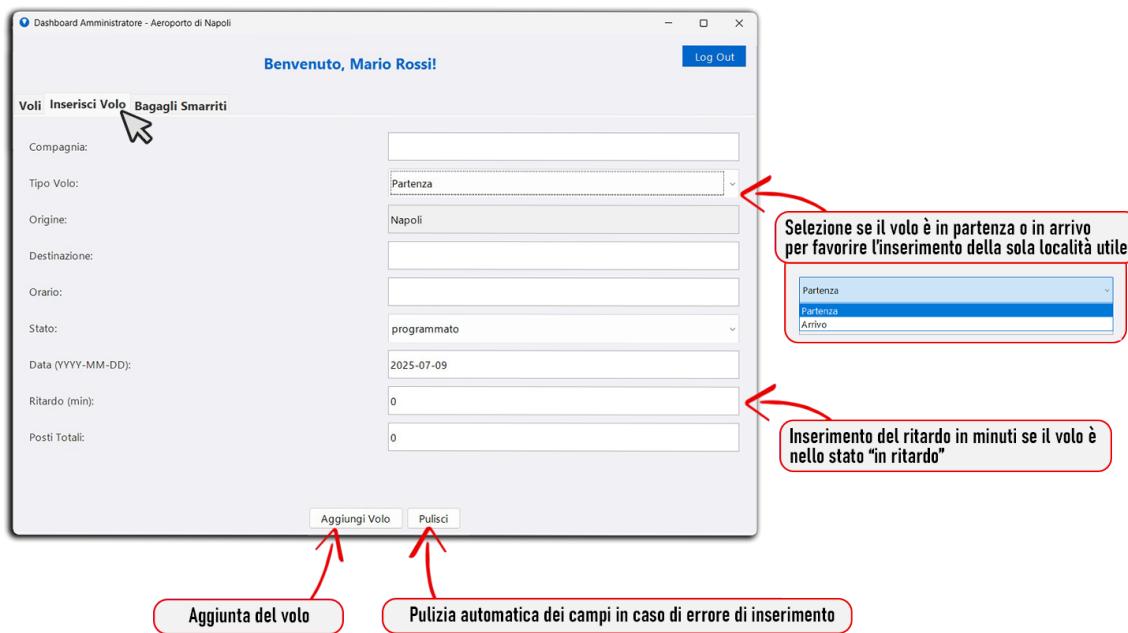
5.2.4 Interfaccia modifica prenotazioni

L'interfaccia per la modifica di una prenotazione consente all'amministratore di aggiornare i dati relativi allo stato della prenotazione e al numero di bagagli associati. Nella scheda secondaria "Bagagli" è possibile gestire ogni singolo bagaglio, aggiornandone lo stato operativo (es. caricato, smarrito). Questa funzionalità è essenziale per il supporto alle operazioni di controllo, carico e smistamento, oltre che per il tracciamento in caso di anomalie.



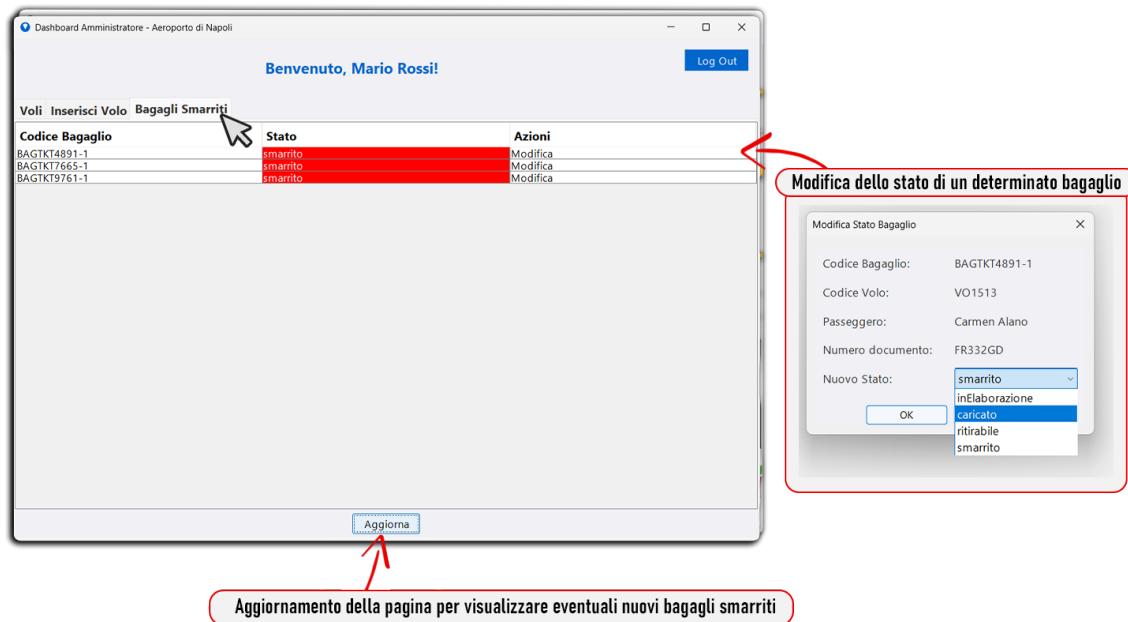
5.2.5 Interfaccia inserimento volo

La sezione "Inserisci Volo" permette agli amministratori di aggiungere nuovi voli al sistema, specificando tutti i parametri richiesti: compagnia, tratta, data, orario, stato, eventuale ritardo e numero di posti disponibili. Un apposito menu consente di distinguere tra voli in partenza e in arrivo, facilitando l'inserimento corretto delle località di origine e destinazione. In caso di errore, il sistema prevede una funzione per la pulizia automatica dei campi



5.2.6 Interfaccia bagagli smarriti

L'interfaccia dedicata alla gestione dei bagagli smarriti consente agli amministratori di monitorare le segnalazioni effettuate dagli utenti e di aggiornare manualmente lo stato di ciascun bagaglio. Per ogni elemento vengono mostrati codice bagaglio, volo associato, passeggero e documento identificativo. È possibile modificare lo stato selezionando uno dei valori disponibili (smarrito, in elaborazione, caricato, ritirabile), al fine di garantire un tracciamento puntuale e affidabile.



5.3 Utente

5.3.1 Interfaccia Visualizzazione e ricerca voli

L'interfaccia "Home" fornisce agli utenti una panoramica aggiornata di tutti i voli presenti nel sistema, visualizzandone codice, compagnia, origine, destinazione, orario, stato, data e ritardo previsto. È possibile effettuare ricerche tramite diversi parametri (data, compagnia, codice volo, aeroporto di partenza o di arrivo) grazie a un sistema di filtraggio avanzato. L'interfaccia è adattabile e scorrevole, garantendo una visualizzazione chiara anche in presenza di numerosi voli.

Pulsante per uscire dal pannello e tornare alla pagina di accesso

Filtro per la ricerca di voli tramite parametri da poter digitare nella barra delle ricerche

Benvenuto, Pino Esposito!

Ricerca Voli

Cerca per: Data [] Cerca []

Codice Volo	Compagnia	Origine	Destinazione	Orario	Stato	Data	Ritardo (min)
IT7158	ITA Airways	Napoli	Malta	23:00:00	programmato	2025-07-07	5
EA6706	EasyJet	Napoli	Catania	18:45:00	programmato	2025-07-07	5
VO3853	Volotea	Tenerife Sud	Napoli	13:00:00	programmato	2025-07-07	5
IT5560	ITA Airways	Roma	Napoli	14:15:00	programmato	2025-07-07	30
W16609	Wizz Air	Budapest	Napoli	19:00:00	programmato	2025-07-07	15
RY7886	Ryanair	Napoli	Malaga	06:15:00	programmato	2025-07-07	0
IT6496	ITA Airways	Napoli	Madrid	09:20:00	programmato	2025-07-07	0
RY6287	Ryanair	Napoli	Valencia	18:30:00	programmato	2025-07-07	5
W19579	Wizz Air	Parigi	Napoli	01:00:00	programmato	2025-07-07	25
W19529	Wizz Air	Napoli	Chania	20:15:00	programmato	2025-07-07	0
RY7853	Ryanair	Milano	Napoli	12:15:00	programmato	2025-07-07	0
IT2280	ITA Airways	Atene	Napoli	04:15:00	cancellato	2025-07-07	45

Possibilità di scorrere per una rapida e comoda visualizzazione grazie anche alla finestra adattabile alla dimensione dello schermo

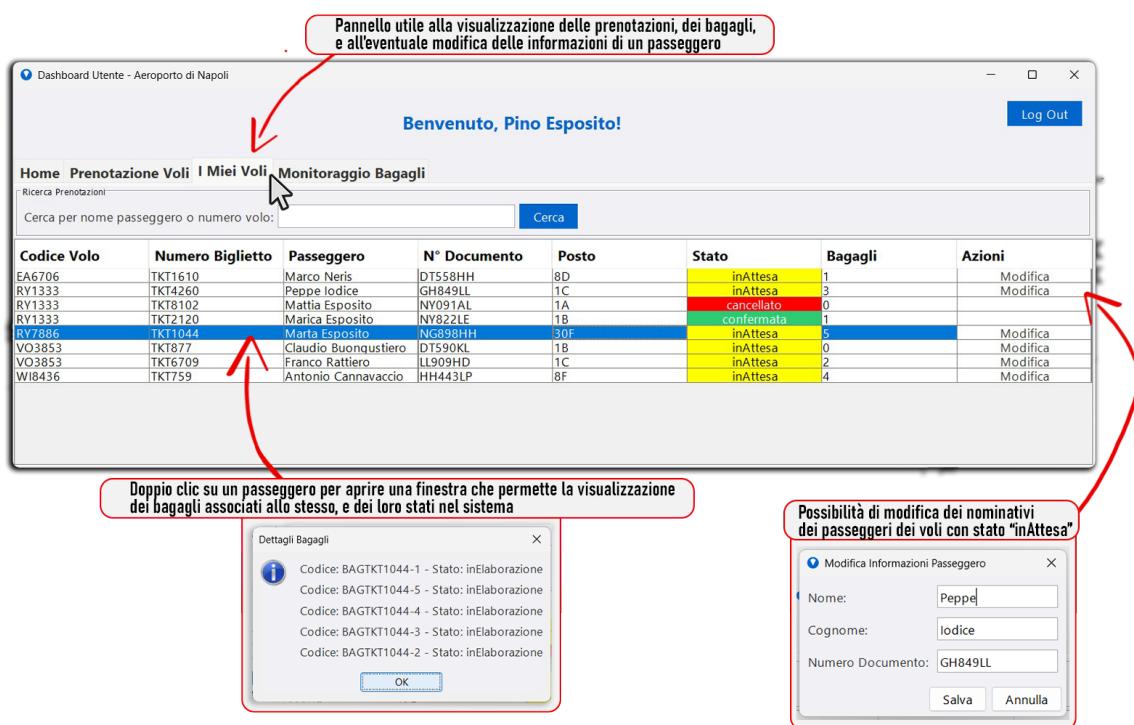
5.3.2 Interfaccia prenotazione voli

L'interfaccia "Prenotazione Voli" permette all'utente di effettuare la prenotazione per uno dei voli disponibili. Dopo aver selezionato il volo desiderato da un menu a tendina, l'utente inserisce i dati anagrafici del passeggero (che può essere diverso dall'intestatario dell'account) e specifica il numero di bagagli da imbarcare, fino a un massimo di cinque. Una volta completata l'immissione dei dati, la prenotazione viene inviata al sistema tramite il pulsante dedicato.



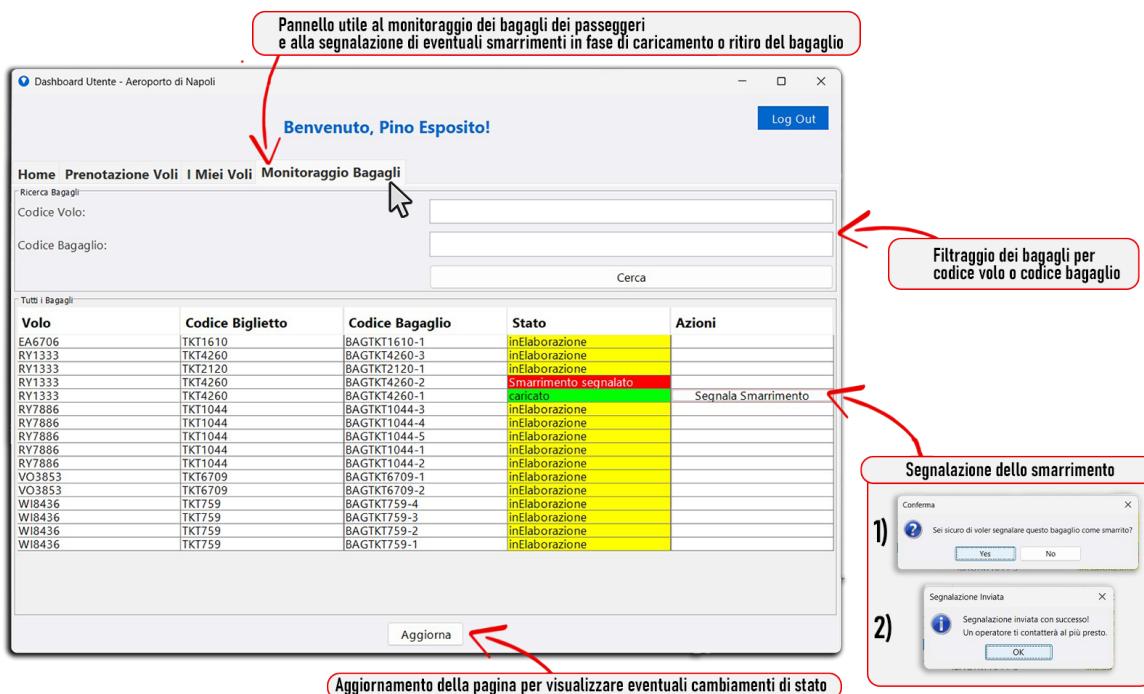
5.3.3 Interfaccia gestione prenotazioni effettuate

Il pannello "I Miei Voli" consente all'utente di visualizzare tutte le prenotazioni associate al proprio profilo, includendo dettagli come numero di biglietto, documento, stato della prenotazione e numero di bagagli. Attraverso un doppio clic su una prenotazione, l'utente può accedere ai dettagli dei bagagli corrispondenti e verificare i rispettivi stati. Inoltre, per i voli ancora in attesa, è possibile modificare le informazioni del passeggero, garantendo così flessibilità e controllo sulla gestione dei dati personali.



5.3.4 Interfaccia monitoraggio bagagli

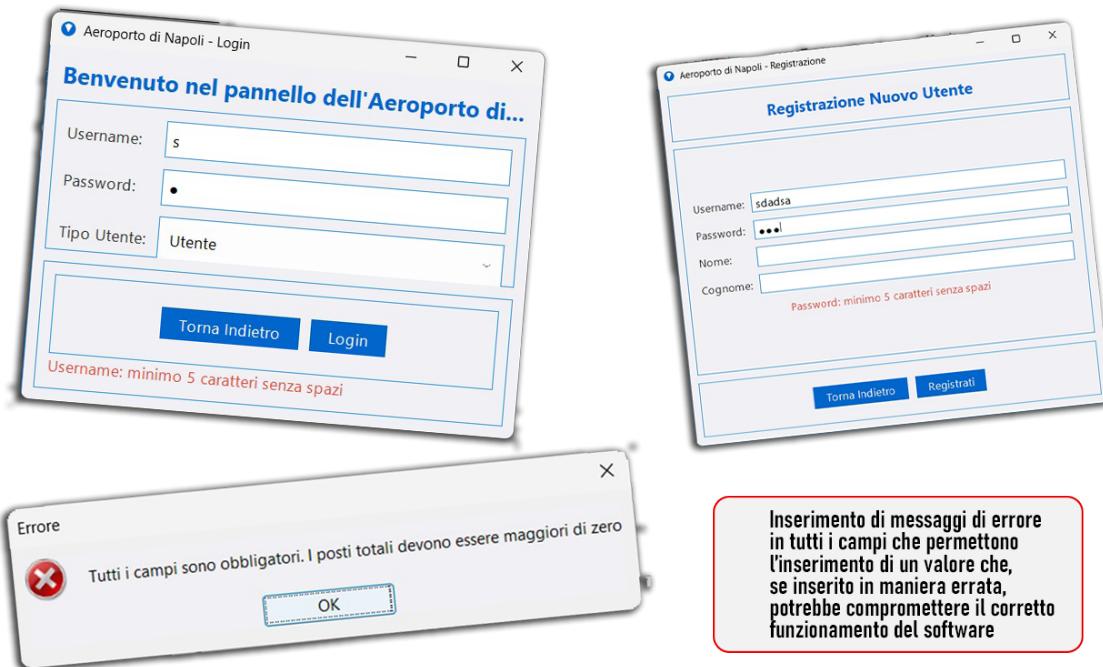
L'interfaccia "Monitoraggio Bagagli" consente all'utente autenticato di visualizzare lo stato aggiornato dei bagagli associati ai propri voli. È possibile effettuare ricerche filtrando per codice volo o codice bagaglio e segnalare eventuali smarrimenti tramite un'apposita funzione. Il sistema mostra un messaggio di conferma per ogni segnalazione effettuata, garantendo così la tracciabilità delle richieste. Il pannello può essere aggiornato manualmente per riflettere eventuali cambiamenti di stato in tempo reale.



Capitolo 6

Gestione errori

Al fine di garantire l'affidabilità e la coerenza dei dati inseriti, l'applicazione implementa un sistema di validazione preventiva sui campi di input presenti nelle schermate di login, registrazione e inserimento dati. In particolare, l'interfaccia fornisce messaggi di errore informativi e contestuali nel caso in cui l'utente inserisca valori non conformi ai vincoli richiesti (ad esempio: username o password troppo brevi, campi obbligatori vuoti, valori numerici non validi). Questa logica di controllo consente di intercettare eventuali anomalie in fase di input, evitando comportamenti imprevisti o compromissioni del flusso applicativo. I messaggi sono mostrati in tempo reale o mediante finestre modali, contribuendo a migliorare l'usabilità e la robustezza del sistema.





UNIVERSITÀ DEGLI STUDI
DI NAPOLI FEDERICO II

Documentazione Progetto Object Orientation

Carmine Sgariglia (N86005069)

Mattia Lemma (N86005170)

Massimo Russo (N86005016)

Anno accademico 2024/2025

Grazie per la cortese attenzione