EMPLOYEE

Object Design Document

Programma Lorenzo

Bonura Davide

Campisi Lucio

Lo Valvo Dario

INDICE

Introduzione

Object design trade-offs

Per la realizzazione del sistema è stato scelto un approccio modulare in modo da favorire la scalabilità, la facilità di implementazione e la gestione di eventuali problematiche.
Nello specifico è stata adoperata un'architettura di tipo Repository, che permette di disaccoppiare in sottosistemi, indipendenti fra di loro, il sistema, i quali comunicano soltanto con il sottosistema di storage. I vari sottosistemi sono composti da un'interfaccia utente che comunica solo con il controllore sottostante che si occupa di gestire le richieste rivolte alla DBMS Boundary, che gestisce le comunicazioni con il sottosistema di storage.

Linee guida per la documentazione del DBMS

Per quanto riguarda il DBMS, si è scelto di utilizzare un modello relazionale gestito tramite MySQLWorkbench. MySQL Workbench è uno strumento visuale di progettazione per database, che integra sviluppo SQL, gestione, modellazione dati, creazione e manutenzione di database MySQL all'interno di un unico ambiente.
Per la comunicazione tra il Sistema e il DBMS si è scelto di usare JDBC (Java DataBase Connectivity). JDBC è un connettore e un driver per database che consente l'accesso e la gestione della persistenza dei dati sulle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DBMS utilizzato.

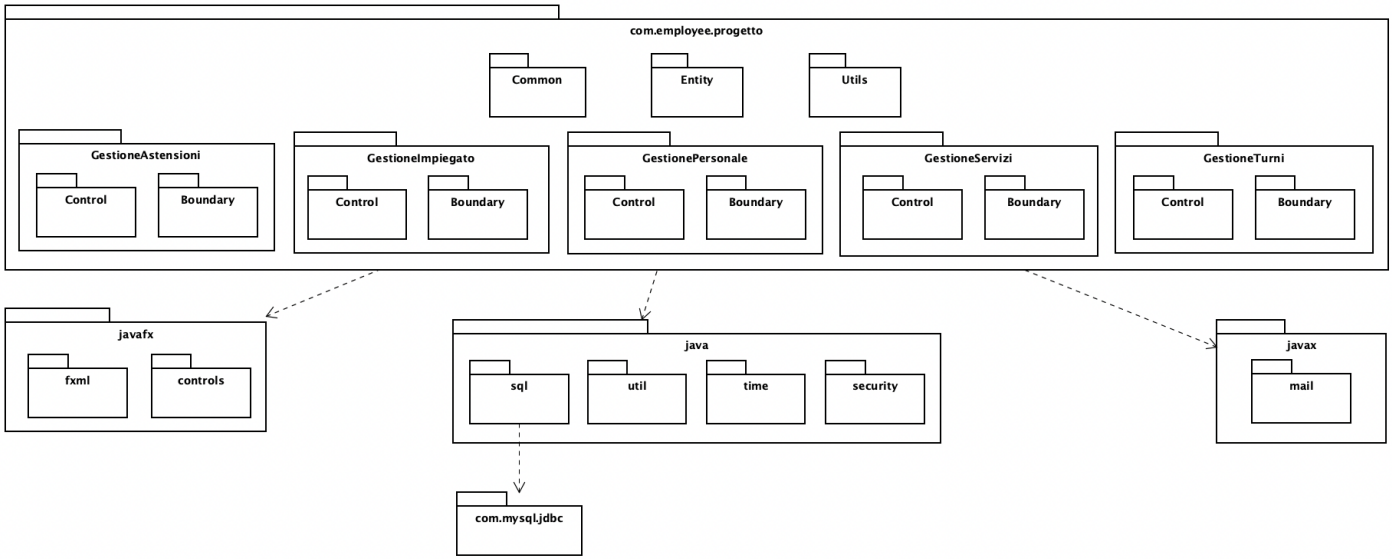
Linee guida per la documentazione dell'invio Mail

Per effettuare l'invio delle mail utilizziamo le API JavaMail e il server SMTP Gmail. È presente anche una funzione che permette di verificare, tramite l'utilizzo di API MailBoxValidator, l'esistenza dell'email inserita in fase di registrazione.

Linee guida per la documentazione delle interfacce

Per quanto riguarda le interfacce grafiche è stato scelto di utilizzare il pacchetto di librerie JavaFX, che ci ha permesso di realizzare le interfacce grafiche tramite documenti di tipo FXML. JavaFX è un pacchetto di librerie component-based, ciò permette quindi di avere una grande riusabilità del codice, oltre a una maggiore leggibilità in fase di scrittura.
Nello specifico, per realizzare le interfacce grafiche è stato utilizzato JavaFX Scene Builder, ovvero un tool per la creazione di documenti di tipo FXML che sfrutta un sistema di drag-and-drop.

Packages



com.employee.progetto

Il package principale del sistema.

Di seguito vi sono le descrizioni dei sottopackage:

- **Common:** Contiene classi che vengono utilizzate nei diversi sottosistemi di GestioneAstensioni, GestioneImpiegato, GestionePersonale, GestioneServizi e GestioneTurni.
- **Entity:** Contiene le classi che modellano le entità di cui è necessaria la rappresentazione nel sistema.
- **Utils:** Contiene le classi utili per la gestione del database, per l'invio delle mail, per la crittografia delle password e per la gestione delle interfacce.
- **GestioneAstensioni:** Contiene le classi utili per la gestione delle astensioni: in particolare Boundary contiene le classi associate alle interfacce realizzate in FXML mentre Control contiene le classi che gestiscono la logica del programma che permette di effettuare le richieste di astensione da parte dell'utente.
- **GestioneImpiegato:** Contiene le classi utili per la gestione dell'impiegato: in particolare Boundary contiene le classi associate alle interfacce realizzate in FXML mentre Control contiene le classi che gestiscono la logica del programma che permette all'impiegato di effettuare funzionalità come visualizzare i propri dati, comunicare un ritardo, richiedere un premesso ecc...
- **GestionePersonale:** Contiene le classi utili per la gestione del personale dell'azienda: in particolare Boundary contiene le classi associate alle interfacce realizzate in FXML mentre Control contiene le classi che gestiscono la logica del programma che permette al personale di effettuare funzionalità come effettuare il login, rilevare la presenza o l'uscita, ecc...

- **GestioneServizi:** Contiene le classi utili per la gestione dei servizi dell'azienda: in particolare Boundary contiene le classi associate alle interfacce realizzate in FXML mentre Control contiene le classi che gestiscono la logica del programma che permette agli utenti di visualizzare i servizi e permette al sistema di controllare che il servizio a priorità più alta sia svolto per più tempo rispetto agli altri e che in caso di assenza di personale di chiudere il servizio a priorità più bassa.
- **GestioneTurni:** Contiene le classi utili per la gestione dei turni dell'azienda: in particolare Boundary contiene le classi associate alle interfacce realizzate in FXML mentre Control contiene le classi che gestiscono la logica del programma che permette agli utenti di visualizzare i turni e permette al sistema di generare i turni ogni 3 mesi.

javafx

Il package è utilizzato per la realizzazione e la gestione delle interfacce grafiche utente.

- **controls:** Definisce il controllo UI (User Interface), i grafici e le skin che sono disponibili per il toolkit JavaFX UI.
- **fxml:** Permette di caricare i file FXML, e gestisce il binding tra le variabili della classe associata e le rispettive componenti grafiche (tramite id ed EventHandler).

java

Contiene i package e le librerie standard di Java.

- **sql:** Il package è utilizzato per gestire le comunicazioni con i DBMS, viene utilizzato nella classe DBMSBoundary per la creazione di connessioni e l'esecuzione di query.
- **util:** Contiene le strutture dati utilizzate per la realizzazione del sistema.
- **time:** Include le API per le date, il tempo, gli istanti e le durate.
- **security:** Include le API per l'implementazione di algoritmi di sicurezza.

javax

Contiene i package e le librerie che estende le librerie standard di Java.

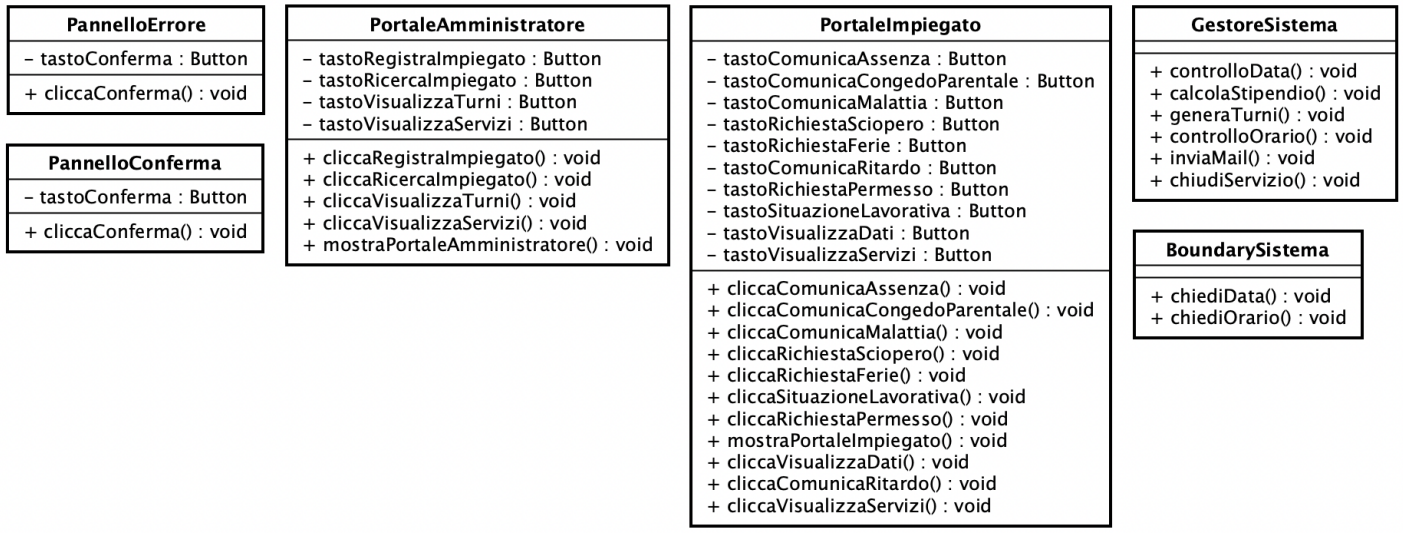
- **mail:** Include l'API JavaMail per l'implementazione delle mail.

com.mysql.jdbc

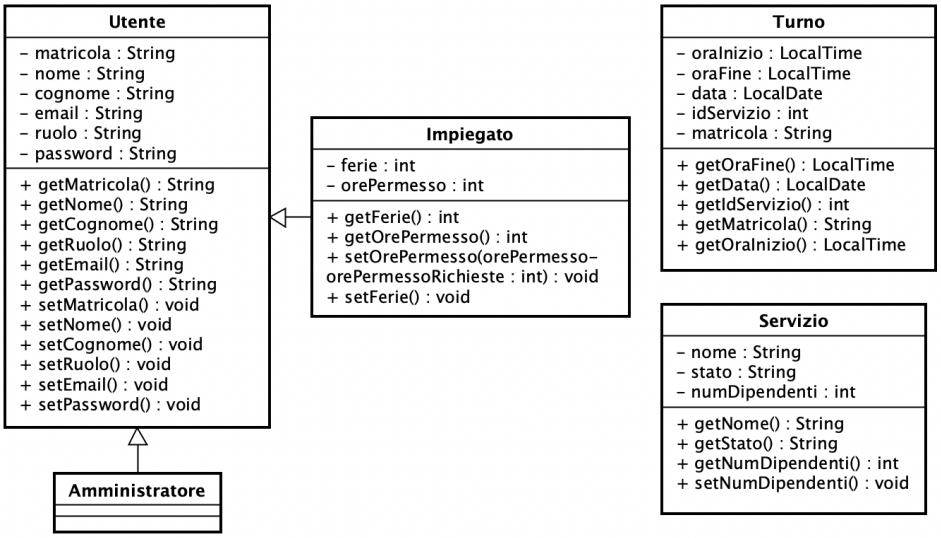
Contiene i driver per le comunicazioni con il DBMS MySQL utilizzato per la realizzazione del sistema.

Object Design UML

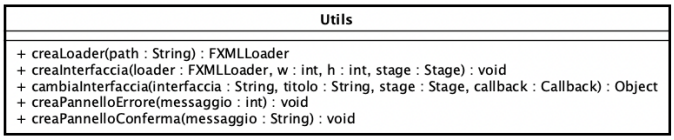
Common

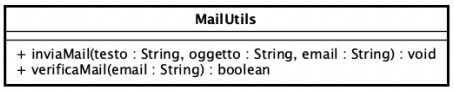


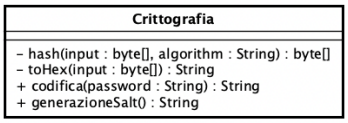
Entity

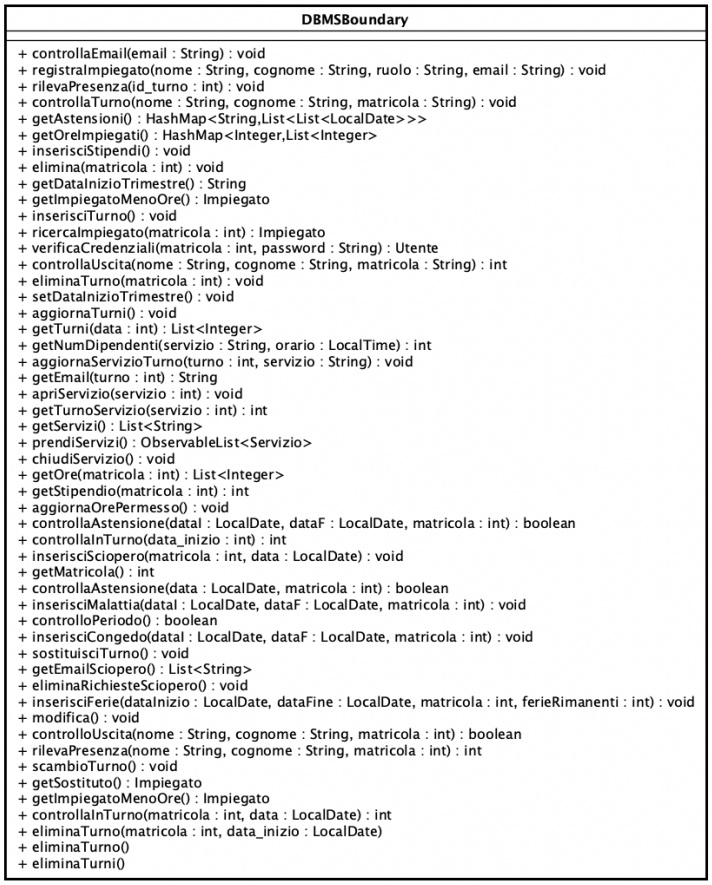


Utils

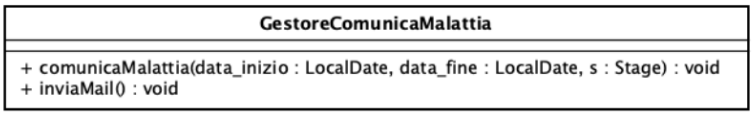


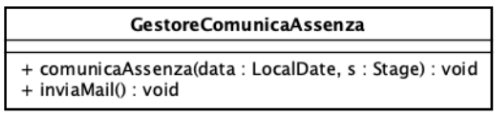


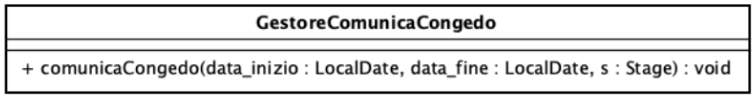


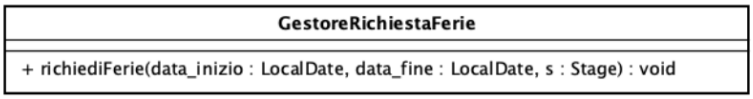


GestioneAstensioni.Control

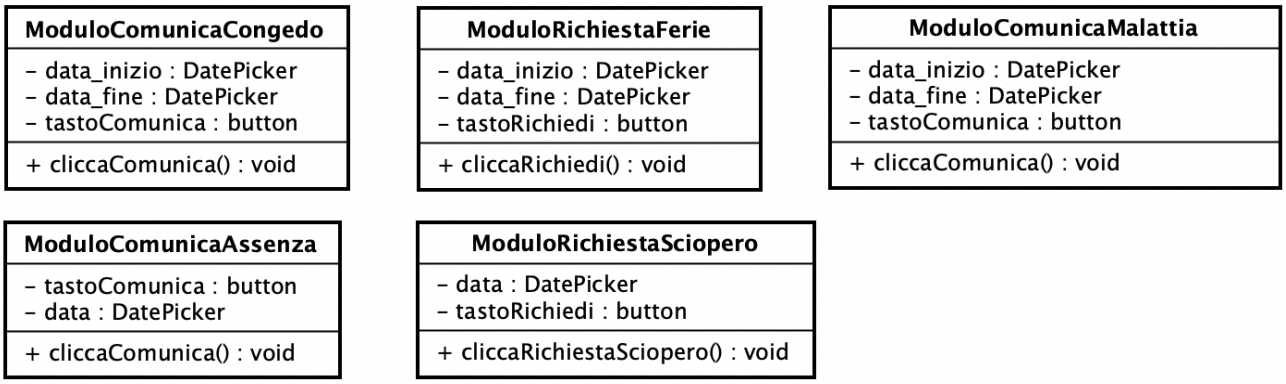




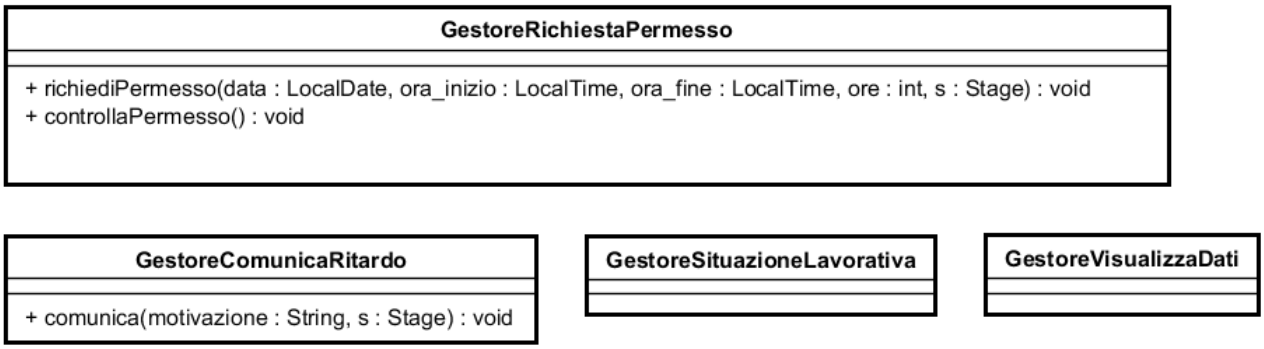




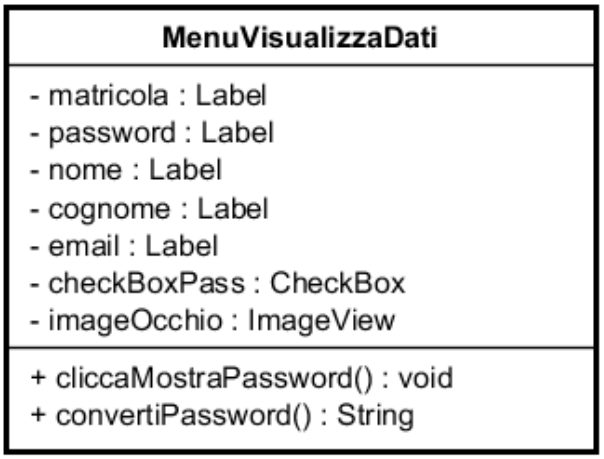
GestioneAstensioni.Boundary

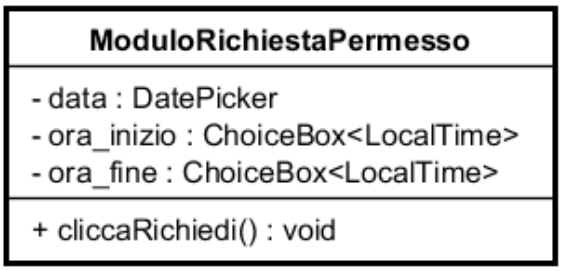


GestioneImpiegato.Control

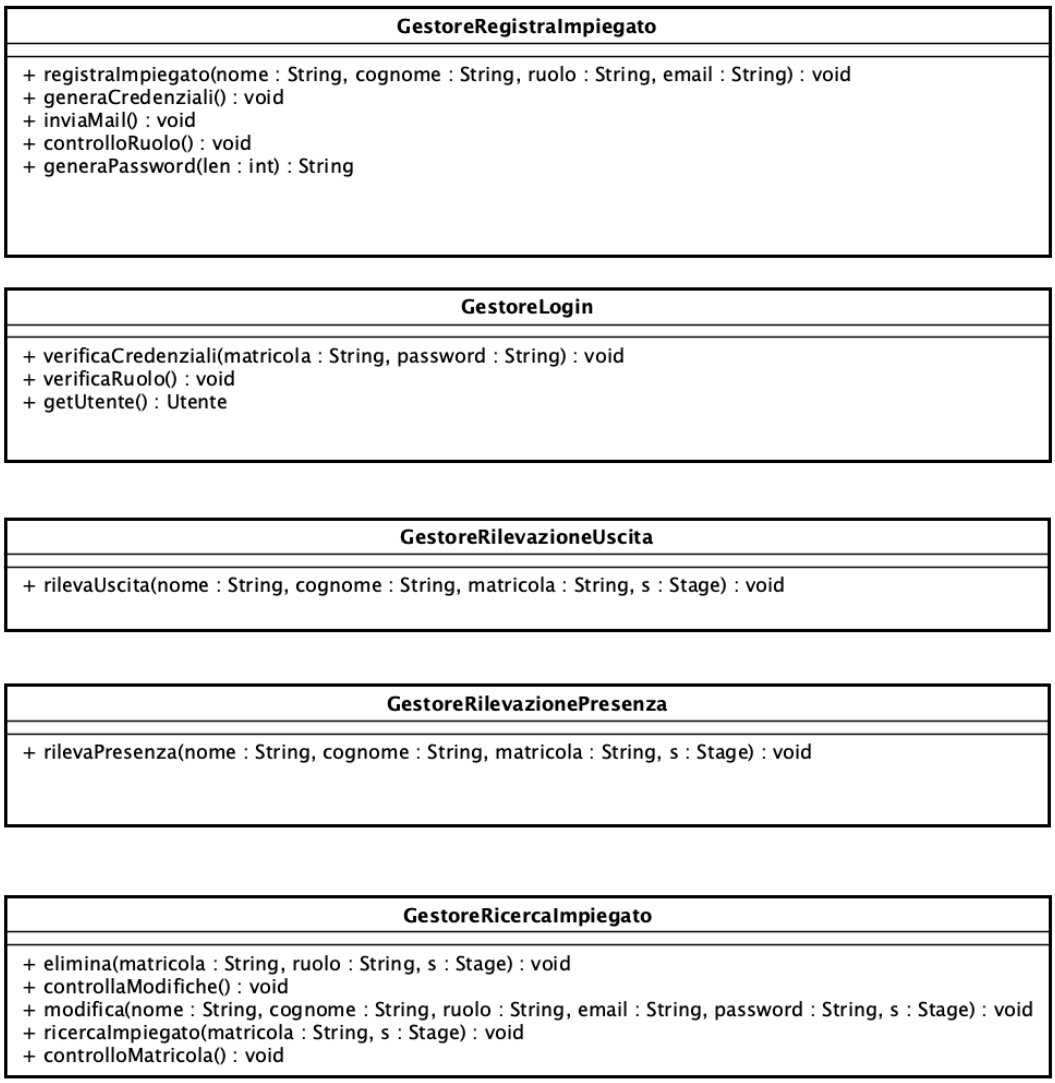


GestioneImpiegato.Boundary

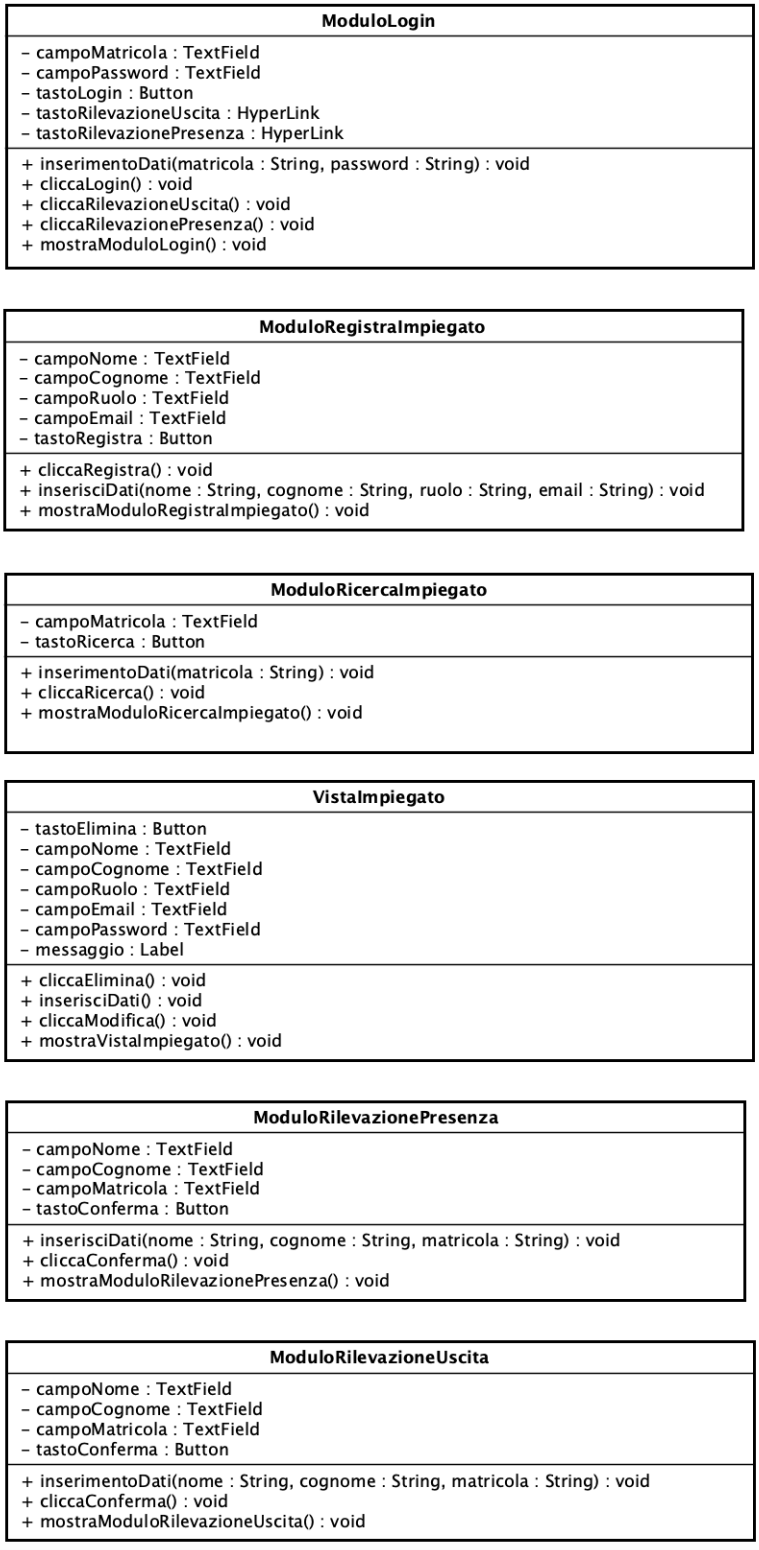




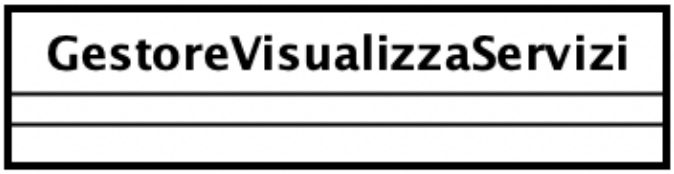
GestionePersonale.Control



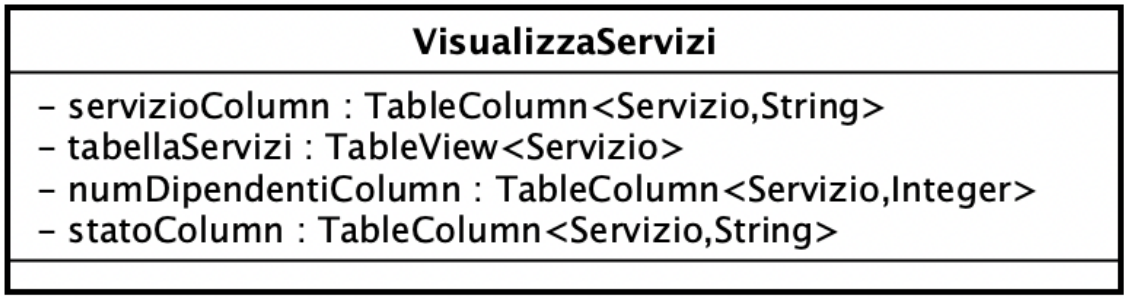
GestionePersonale.Boundary



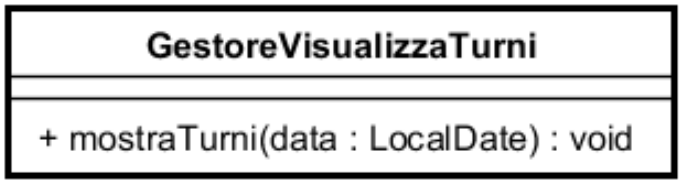
GestioneServizi.Control



GestioneServizi.Boundary



GestioneTurni.Control



GestioneTurni.Boundary