UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO DI INGEGNERIA

ELD

Software Gestionale per Farmacie e Aziende Farmaceutiche

Object Design Document

Studenti:
Daniele Franco
Laura Gioè
Edoardo Terranova

Docente:
Valeria Seidita

Anno Accademico 2021/2022

Sommario

1. Introduzione

1.1 Obiettivi del sistema

L'obiettivo del sistema ELD è la gestione integrata dello stoccaggio e vendita di farmaci di un'azienda farmaceutica a una catena di farmacie, supporta la prenotazione dei farmaci da parte delle farmacie e le consegne.

1.2 Acronimi

ODD	Object Design Document
RAD	Requirement Analysis Document
SDD	System Design Document
UML	Unified Modeling Language
MVC	Model View Controller
DBMS	DataBase Management System
ERD	Entity Relationship Diagram

2. Sistema attuale

Non esiste un software attualmente utilizzato.

3. Sistema proposto

3.1 Object Design Trade-Offs

Per realizzare il sistema è stata utilizzata un'architettura ibrida tra quella Repository e quella Model View Control.

L'architettura Repository garantisce lo scambio di dati tra sottosistemi senza che debbano scambiare messaggi tra loro.

L'architettura MVC suddivide i sottosistemi in componenti logiche che comunicano tra loro:

- **Model**: responsabile di mantenere i dati del sistema, notifica un cambio di stato dei dati alla View.
- **View**: responsabile della visualizzazione dei dati e dell'interazione con l'utente, preleva i dati dalla Model e notifica alla Control quando riceve un input dall'utente.
- **Control**: responsabile della gestione degli input dell'utente, cambia lo stato dei dati della Model e invia dati da visualizzare alla View.

Il sistema è realizzato utilizzando il linguaggio di programmazione Java.

Linee Guida per la Documentazione dei Database

Per la gestione dei Database si è optato per un modello relazionale gestito attraverso MySQL, mentre per permettere la connessione tra DB e sistema si è scelto di usare il Java Database Connectivity (JDBC). Il JDBC è una API per il linguaggio Java che definisce l'accesso ad un Database da parte di un client e assieme alla libreria SQL permette la gestione dei dati persistenti.

Linee Guida per la Documentazione delle Interfacce

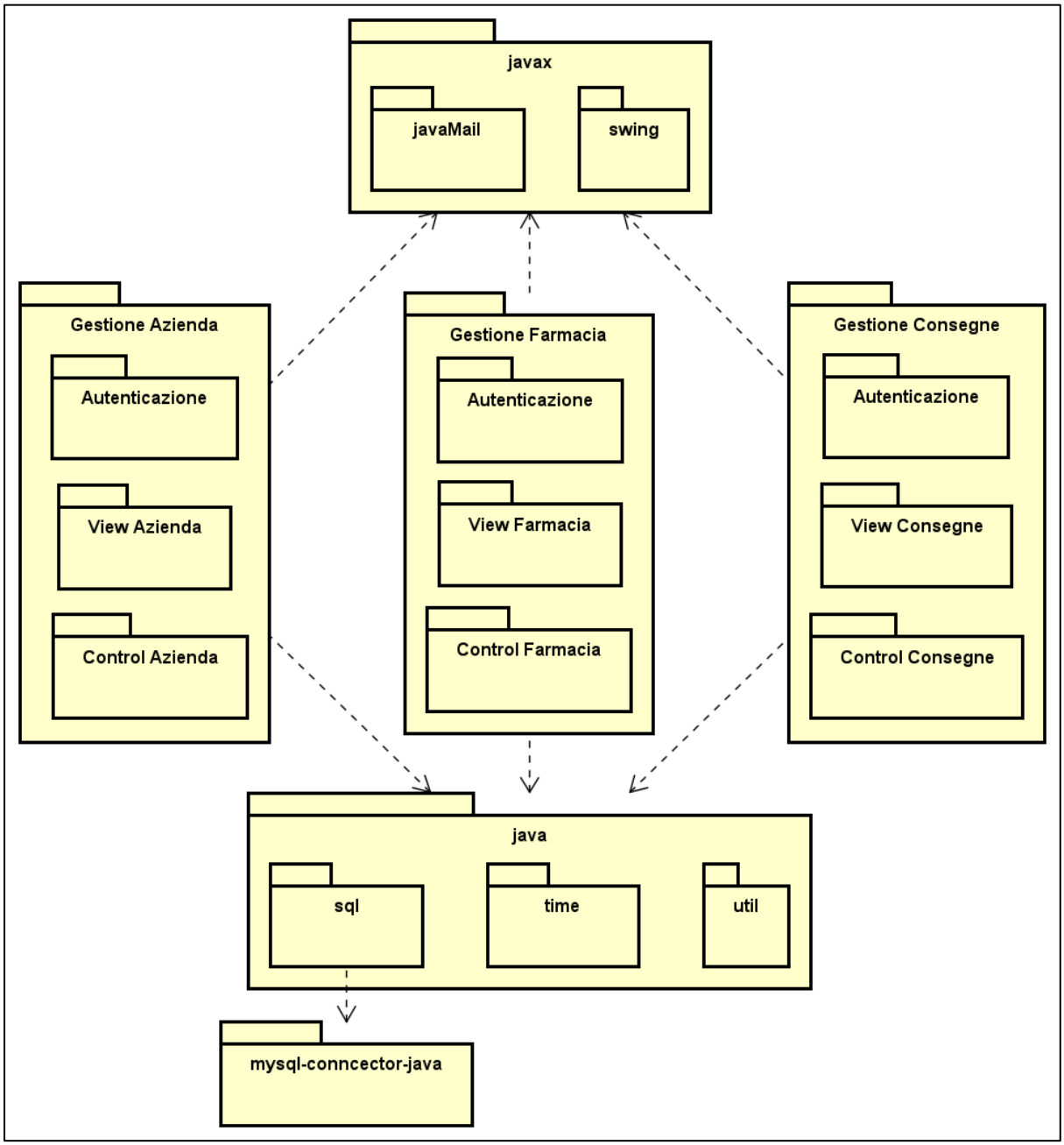
Per lo sviluppo delle interfacce grafiche si è deciso di usufruire della libreria Java Swing, una estensione della libreria Abstract Window Toolkit, in modo da poter usufruire degli oggetti grafici (widget) a disposizione dalla libreria stessa.

La libreria Swing, grazie al suo design pattern MVC, rende i suoi componenti altamente riutilizzabili e l'uso della IDE JavaNetBeans ha reso più efficiente la gestione della Graphic User Interface grazie alla tecnica drag-and-drop per la creazione di widget in codice Java.

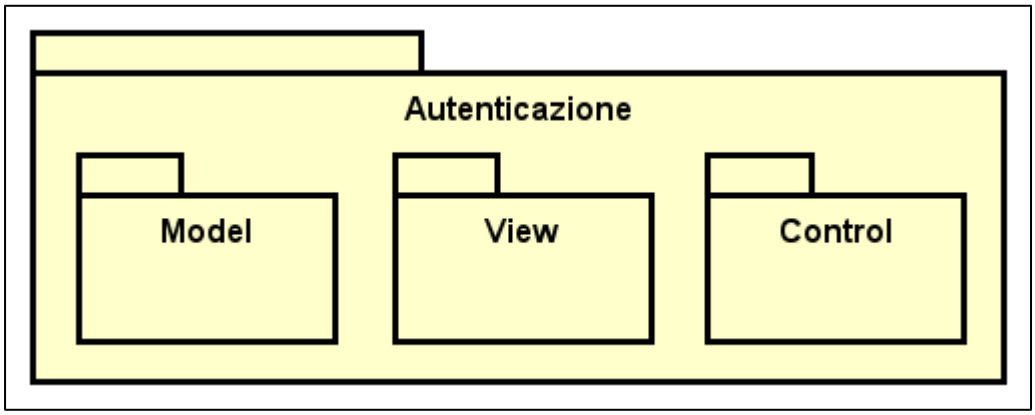
Linee Guida per la Documentazione dell'invio E-Mail

Per inviare e-mail agli utenti per il Recupera Password, si usa l'API JavaMail la quale implementa metodi per gestire le connessioni ai server di posta elettronica. L'indipendenza dalla piattaforma e dai protocolli di posta rende questo framework completo ma facile da gestire e con il metodo send, dopo aver specificato tutte le proprietà ed il corpo della e-mail, invierà i dati alla casella di posta elettronica corretta.

3.2 Package


javax
javaMail
swing
Gestione Azienda
Autenticazione
View Azienda
Control Azienda
Gestione Farmacia
Autenticazione
View Farmacia
Control Farmacia
Gestione Consegne
Autenticazione
View Consegne
Control Consegne
java
sql
time
util
mysql-conncector-java

Autenticazione



Il package Autenticazione contiene le classi, distribuite secondo l'architettura MVC, che permettono all'utente di autenticarsi e di cambiare la password. È comune a tutti gli altri package.

Gestione Azienda

Il package gestione Azienda contiene le classi, distribuite secondo l'architettura MVC, che permettono all'impiegato di eseguire le funzionalità a lui consentite: visualizzare i farmaci in magazzino, gli ordini e le segnalazioni; effettuare ordini per conto di una farmacia; modificare e annullare un ordine; e caricare e scaricare i farmaci.

Gestione Farmacia

Il package Gestione Farmacia contiene tutte le classi, distribuite secondo l'architettura MVC, che permettono al farmacista di eseguire le funzionalità a lui consentite: visualizzare i farmaci in deposito e le prenotazioni; prenotare, caricare e scaricare i farmaci; annullare e modificare una prenotazione; e segnalare le consegne che non rispecchiano la prenotazione.

Gestione Consegne

Il package Gestione Consegne contiene tutte le classi, distribuite secondo l'architettura MVC, che permettono al corriere di eseguire le funzionalità a lui consentite: visualizzare le consegne e farle firmare al farmacista.

java

Il package java contiene le librerie standard del linguaggio Java, tra cui i seguenti package:

- **sql**: contiene le classi che si occupano di stabilire una connessione con il DBMS e gestirne lo scambio di dati attraverso l'esecuzione di query. È utilizzato dalle classi Control
- **time**: contiene le classi che si occupano di gestire le date, l'orario, gli istanti e gli intervalli di tempo. È utilizzato dalle classi Control
- **util**: contiene le classi che gestiscono le strutture dati

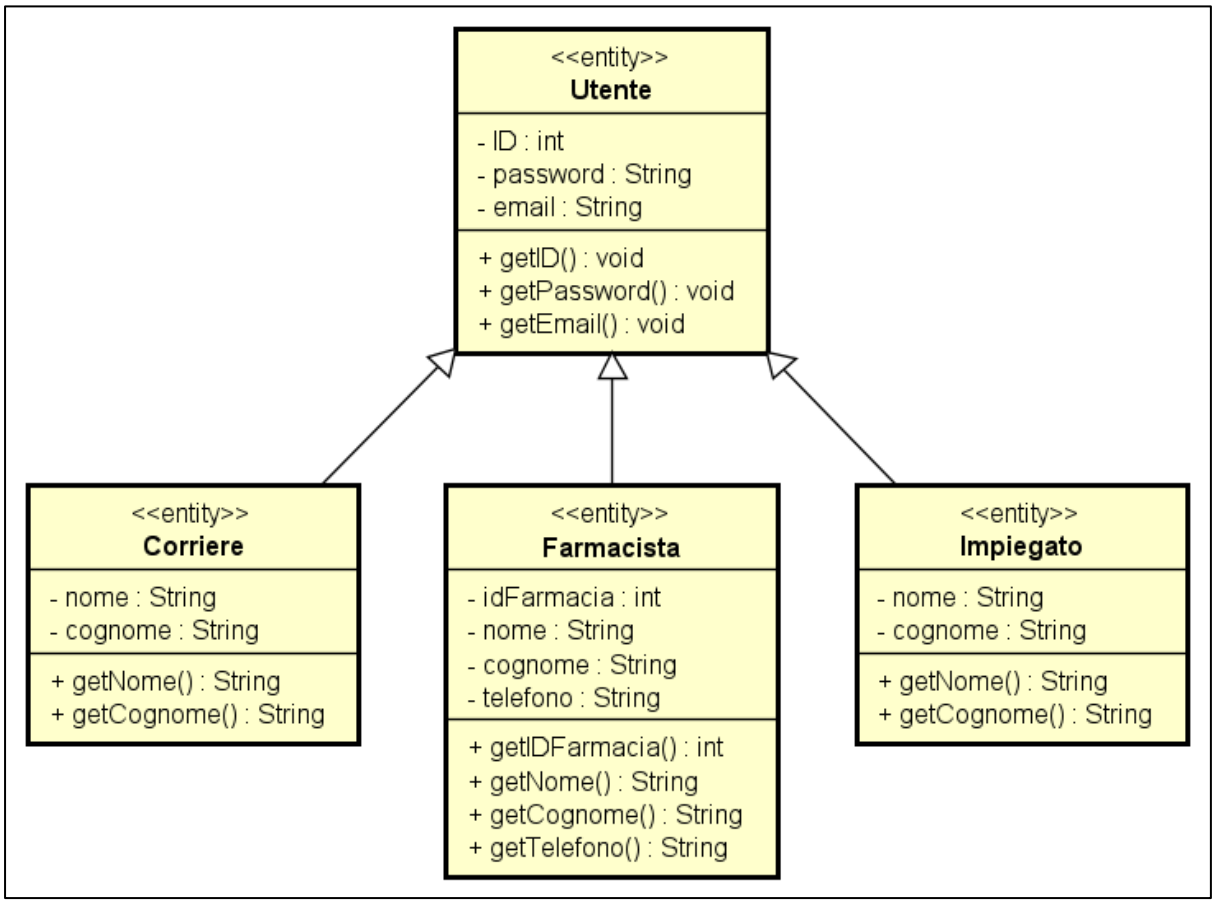
javax

Il package javax contiene pacchetti di estensione del linguaggio Java, tra cui i seguenti package:

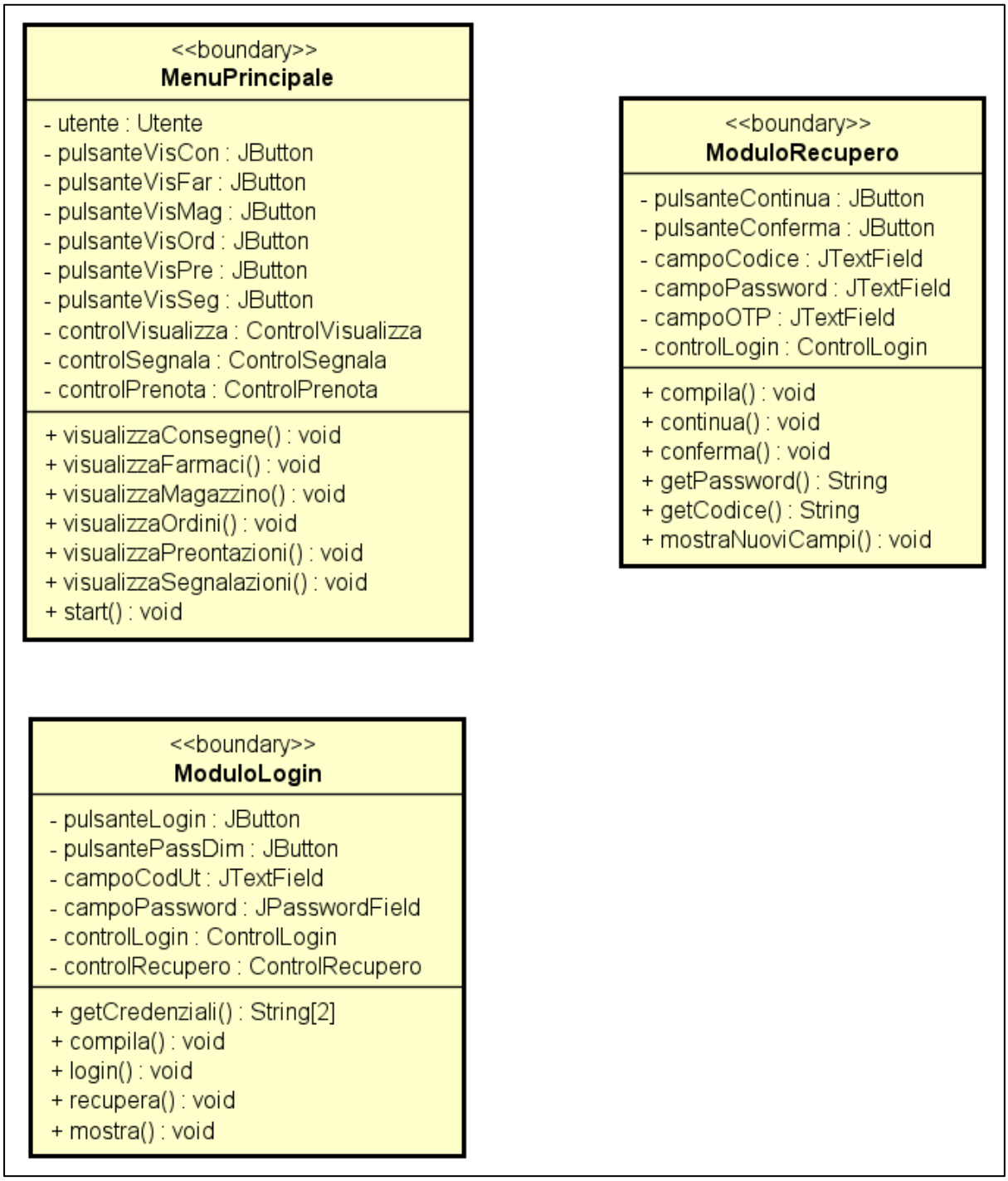
- **swing**: contiene le classi che definiscono le interfacce grafiche e i metodi che gestiscono gli input dell'utente. È utilizzato dalle classi View.
- **javaMail**: contiene le classi che definiscono le connessioni agli indirizzi di posta elettronica e i metodi che gestiscono l'invio delle Mail. È utilizzato dalle classi Control del package Autenticazione.

3.3 Diagramma delle classi

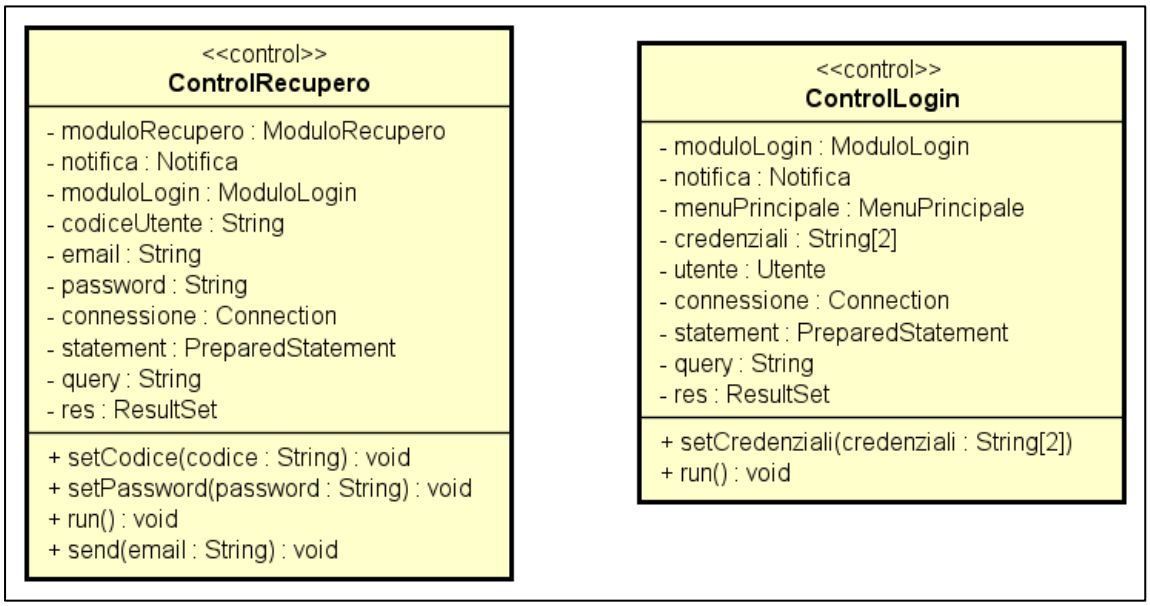
Autenticazione.Model



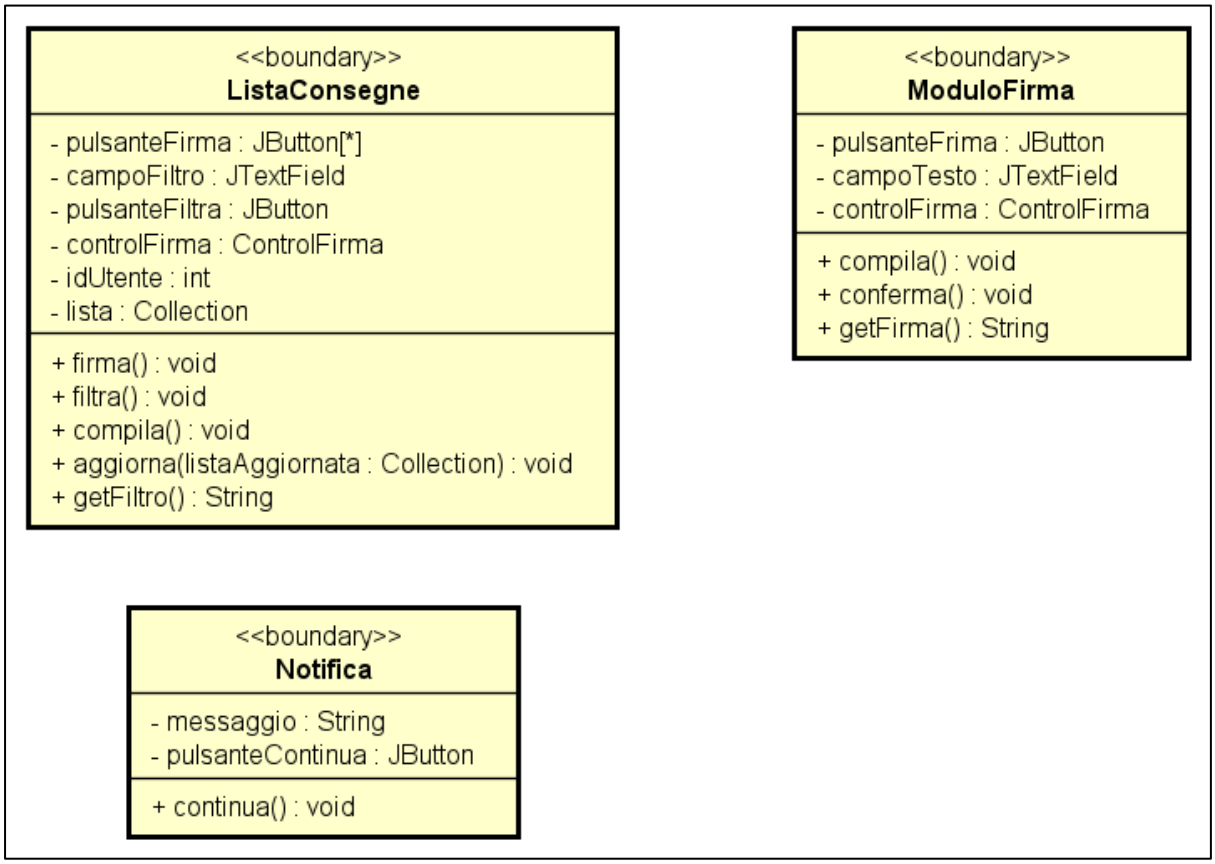
Autenticazione.View



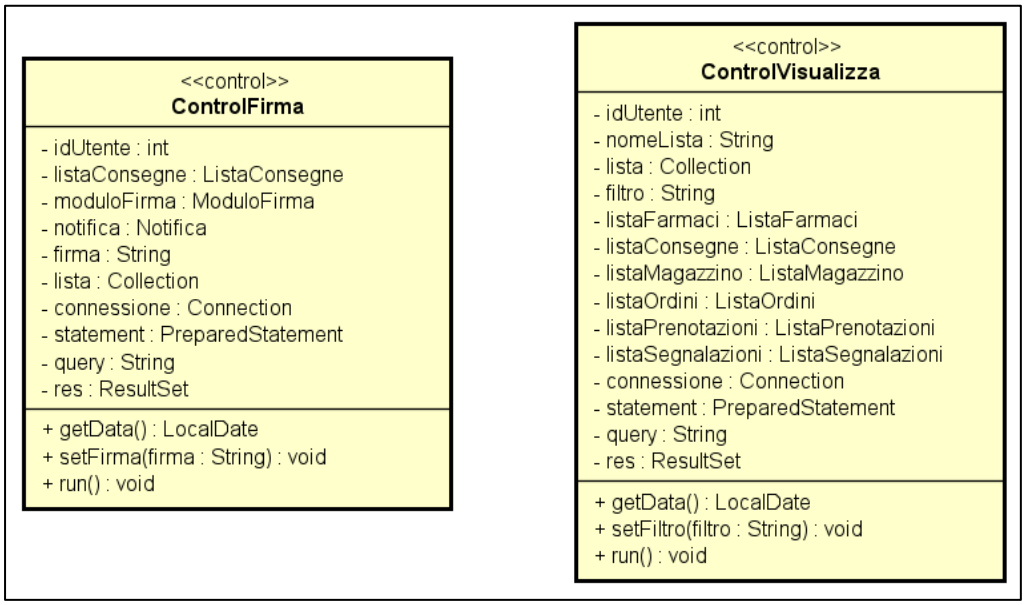
Autenticazione.Control



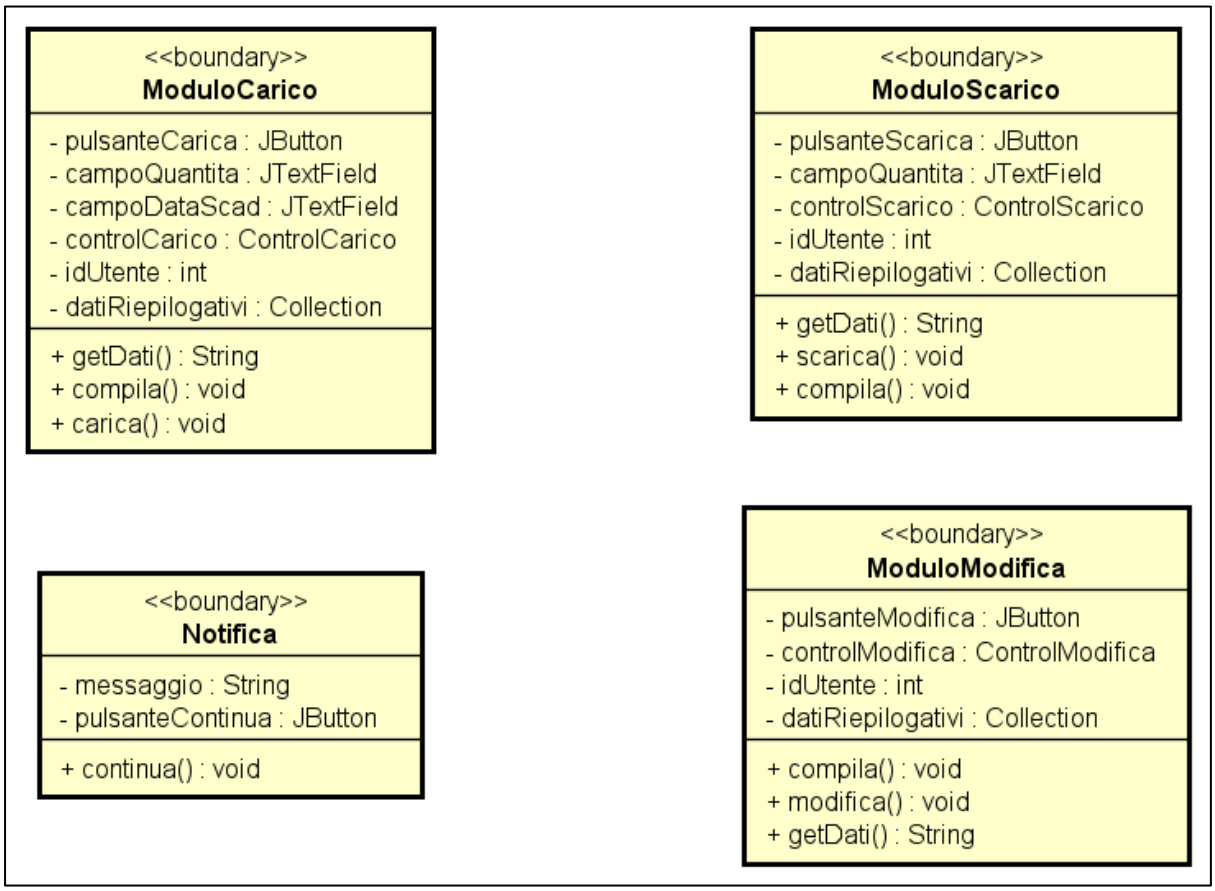
Gestione Consegne.View Consegne

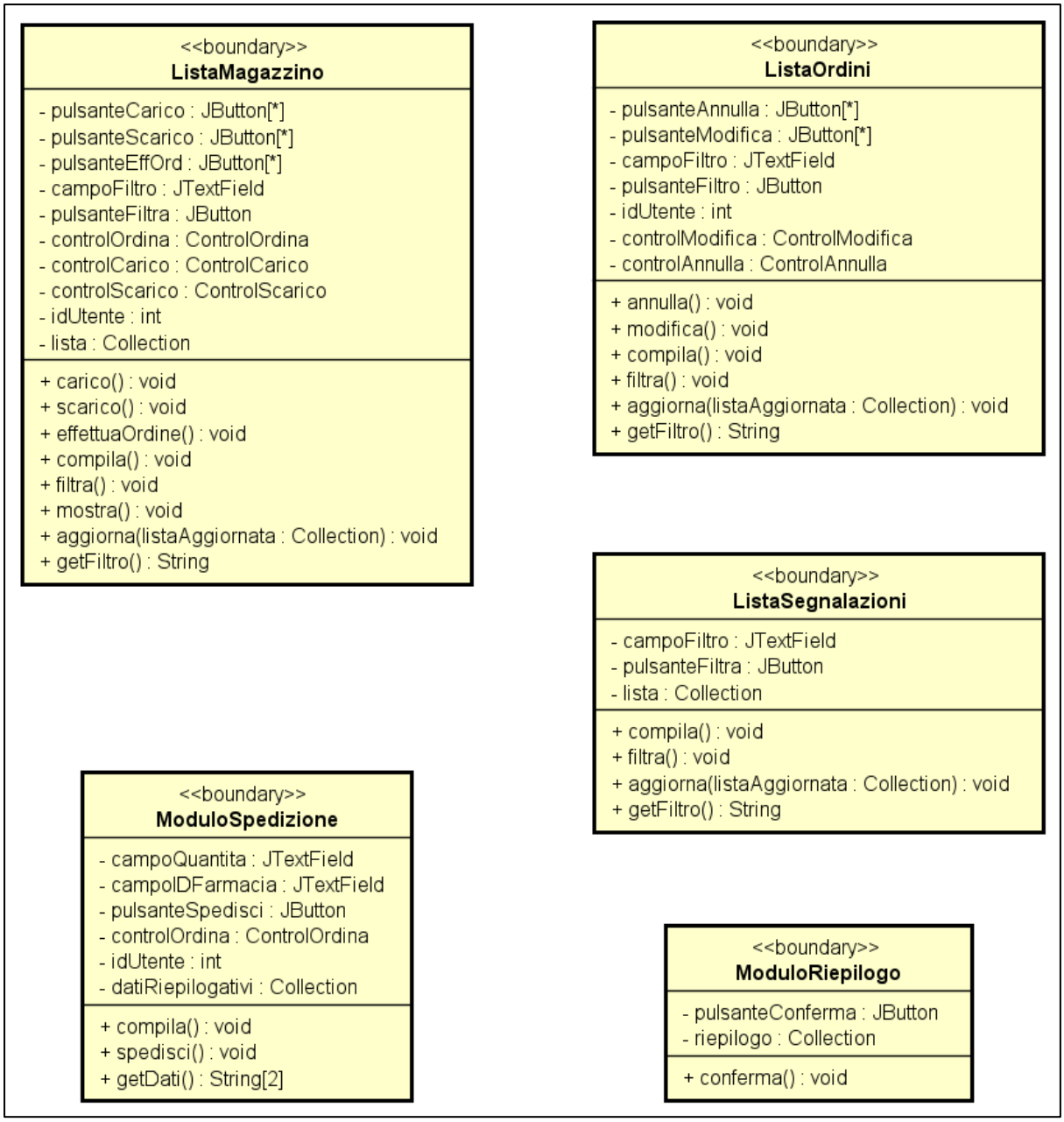


Gestione Consegne.Control Consegne

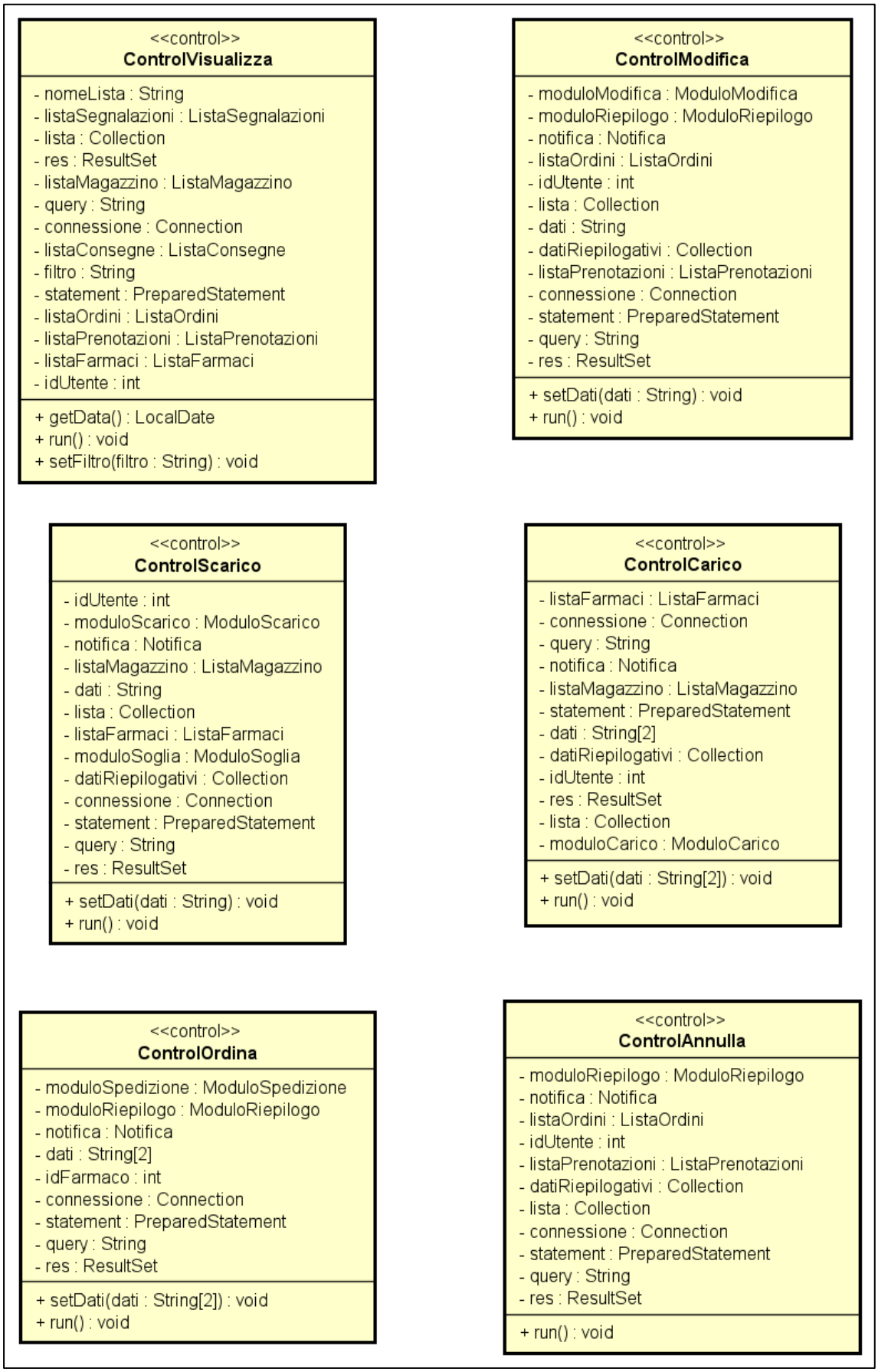


Gestione Azienda.View Azienda

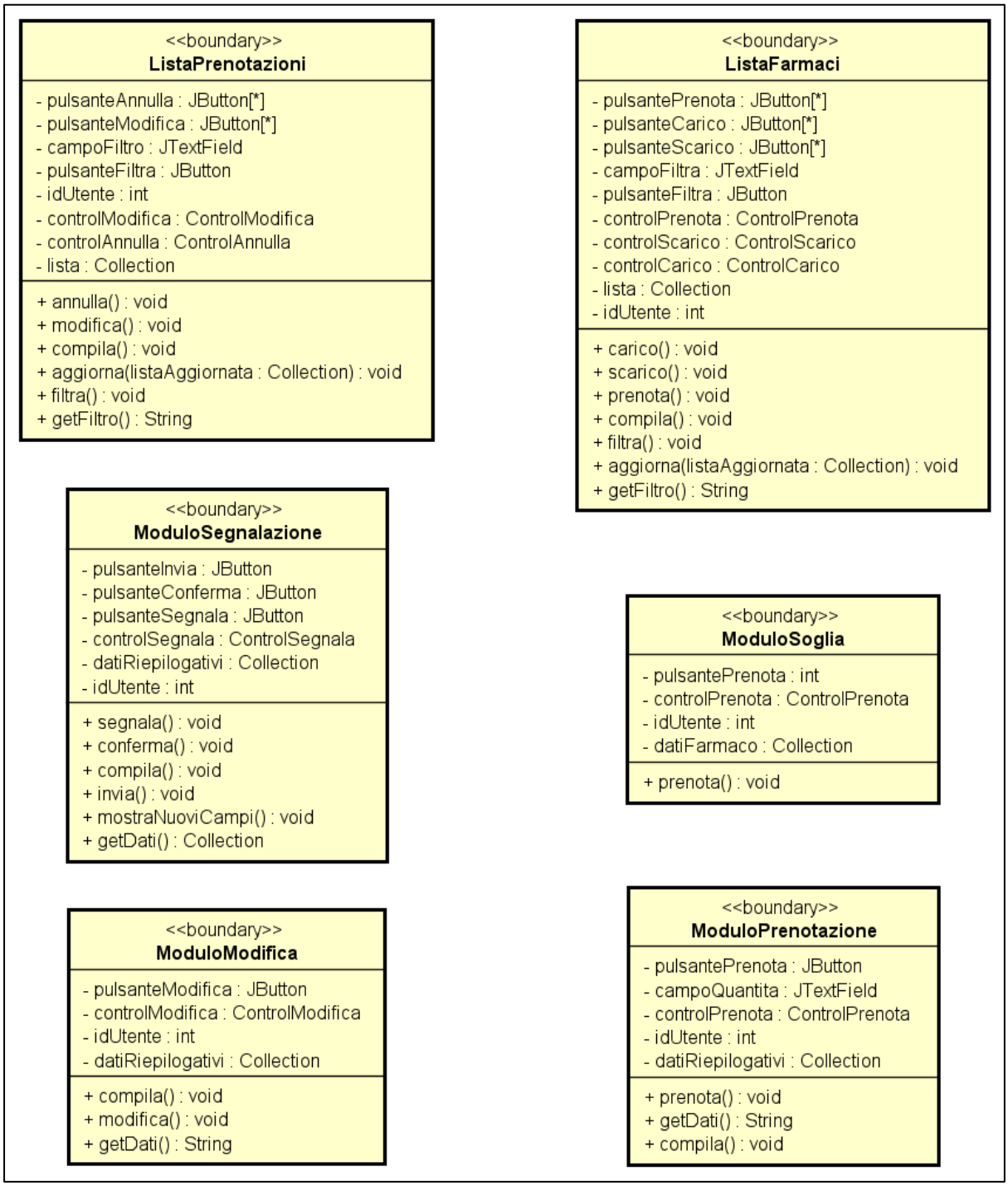


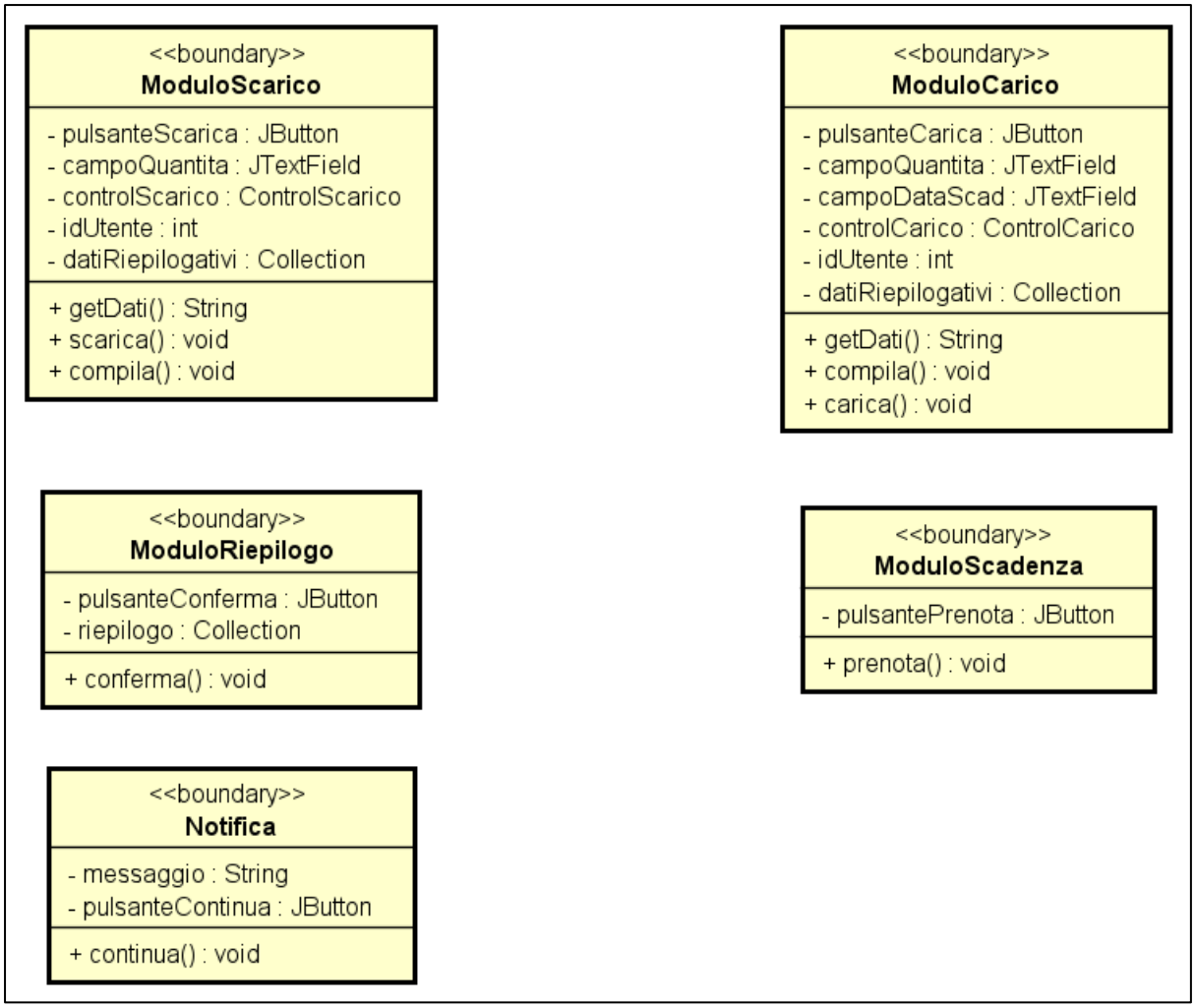

<<boundary>>
ListaMagazzino
- pulsanteCarico : JButton[*]
- pulsanteScarico : JButton[*]
- pulsanteEffOrd : JButton[*]
- campoFiltro : JTextField
- pulsanteFiltra : JButton
- controlOrdina : ControlOrdina
- controlCarico : ControlCarico
- controlScarico : ControlScarico
- idUtente : int
- lista : Collection
+ carico() : void
+ scarico() : void
+ effettuaOrdine() : void
+ compila() : void
+ filtra() : void
+ mostra() : void
+ aggiorna(listaAggiornata : Collection) : void
+ getFiltro() : String
<<boundary>>
ListaOrdini
- pulsanteAnnulla : JButton[*]
- pulsanteModifica : JButton[*]
- campoFiltro : JTextField
- pulsanteFiltro : JButton
- idUtente : int
- controlModifica : ControlModifica
- controlAnnulla : ControlAnnulla
+ annulla() : void
+ modifica() : void
+ compila() : void
+ filtra() : void
+ aggiorna(listaAggiornata : Collection) : void
+ getFiltro() : String
<<boundary>>
ListaSegnalazioni
- campoFiltro : JTextField
- pulsanteFiltra : JButton
- lista : Collection
+ compila() : void
+ filtra() : void
+ aggiorna(listaAggiornata : Collection) : void
+ getFiltro() : String
<<boundary>>
ModuloSpedizione
- campoQuantita : JTextField
- campoIDFarmacia : JTextField
- pulsanteSpedisci : JButton
- controlOrdina : ControlOrdina
- idUtente : int
- datiRiepilogativi : Collection
+ compila() : void
+ spedisci() : void
+ getDati() : String[2]
<<boundary>>
ModuloRiepilogo
- pulsanteConferma : JButton
- riepilogo : Collection
+ conferma() : void

Gestione Azienda.Control Azienda

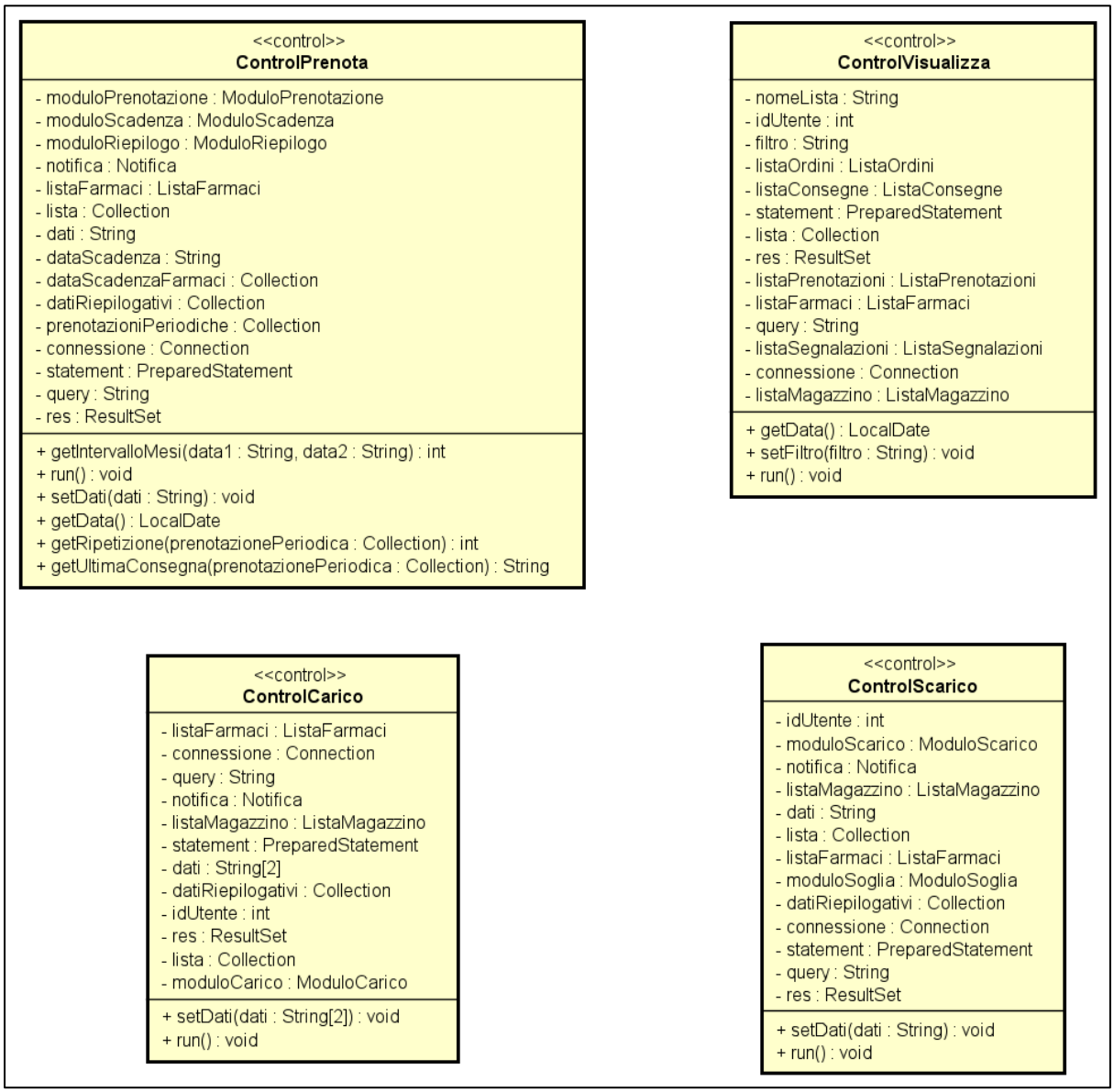


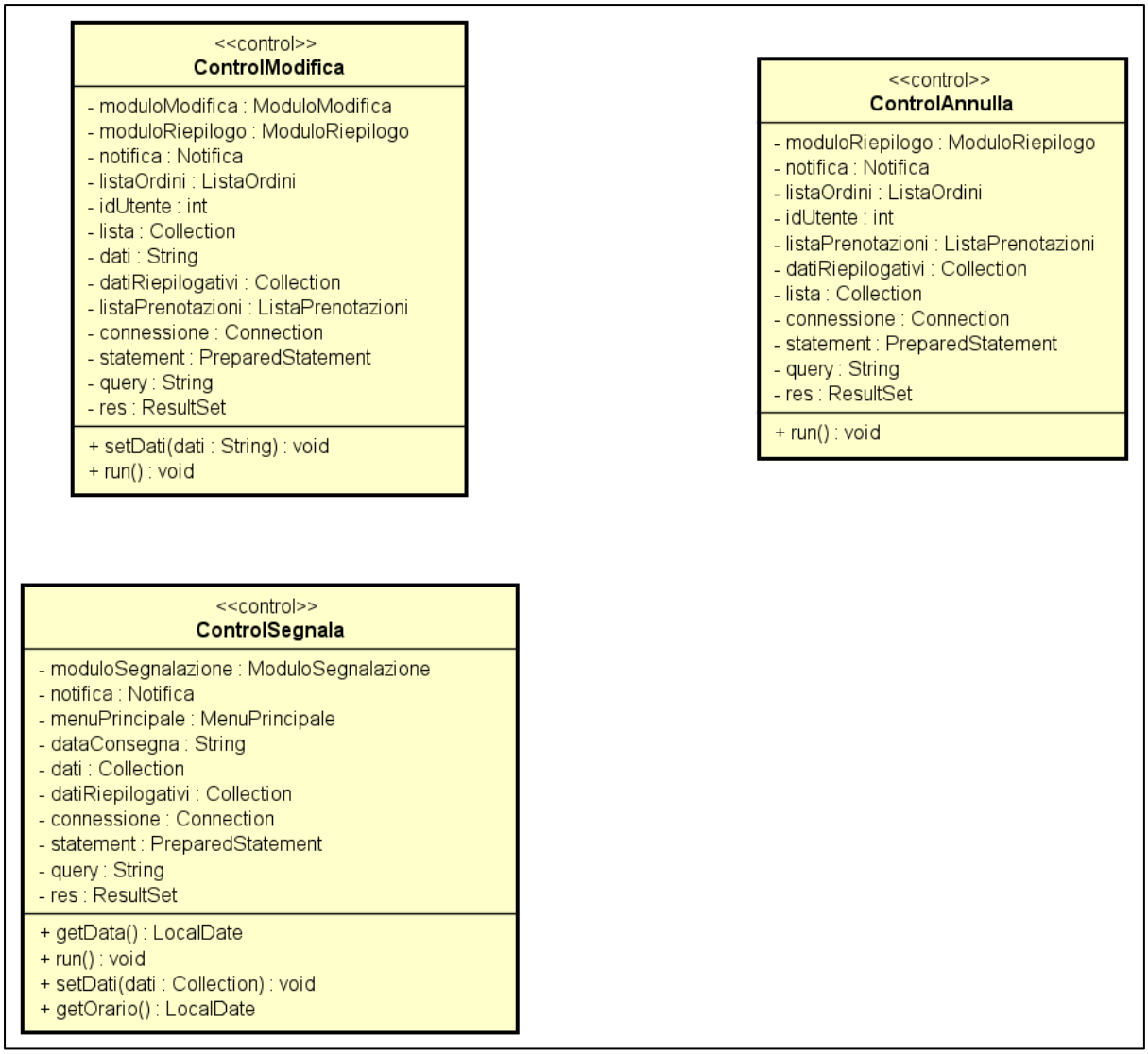
Gestione Farmacia. View Farmacia




<<boundary>>
ModuloScarico
- pulsanteScarica : JButton
- campoQuantita : JTextField
- controlScarico : ControlScarico
- idUtente : int
- datiRiepilogativi : Collection
+ getDati() : String
+ scarica() : void
+ compila() : void
<<boundary>>
ModuloCarico
- pulsanteCarica : JButton
- campoQuantita : JTextField
- campoDataScad : JTextField
- controlCarico : ControlCarico
- idUtente : int
- datiRiepilogativi : Collection
+ getDati() : String
+ compila() : void
+ carica() : void
<<boundary>>
ModuloRiepilogo
- pulsanteConferma : JButton
- riepilogo : Collection
+ conferma() : void
<<boundary>>
ModuloScadenza
- pulsantePrenota : JButton
+ prenota() : void
<<boundary>>
Notifica
- messaggio : String
- pulsanteContinua : JButton
+ continua() : void

Gestione Farmacia.Control Farmacia




<<control>>
ControlModifica
- moduloModifica : ModuloModifica
- moduloRiepilogo : ModuloRiepilogo
- notifica : Notifica
- listaOrdini : ListaOrdini
- idUtente : int
- lista : Collection
- dati : String
- datiRiepilogativi : Collection
- listaPrenotazioni : ListaPrenotazioni
- connessione : Connection
- statement : PreparedStatement
- query : String
- res : ResultSet
+ setDati(dati : String) : void
+ run() : void
<<control>>
ControlAnnulla
- moduloRiepilogo : ModuloRiepilogo
- notifica : Notifica
- listaOrdini : ListaOrdini
- idUtente : int
- listaPrenotazioni : ListaPrenotazioni
- datiRiepilogativi : Collection
- lista : Collection
- connessione : Connection
- statement : PreparedStatement
- query : String
- res : ResultSet
+ run() : void
<<control>>
ControlSegnala
- moduloSegnalazione : ModuloSegnalazione
- notifica : Notifica
- menuPrincipale : MenuPrincipale
- dataConsegna : String
- dati : Collection
- datiRiepilogativi : Collection
- connessione : Connection
- statement : PreparedStatement
- query : String
- res : ResultSet
+ getData() : LocalDate
+ run() : void
+ setDati(dati : Collection) : void
+ getOrario() : LocalDate