

PHARMALINK

Requirements Analysis Document

Team SRV



Davide Sgroi
Francesco Paolo Rosone
Salvatore Viganò

Sommario

Introduzione	2
Scelte di implementazione e linee guida	2
Packages	3
Object Design UML	5
Entity	5
Utils	6
Common	7
GestioneFarmaci	7
GestioneConsegne	8
GestioneOrdine	9
Autenticazione	11

Introduzione

Scelte di implementazione e linee guida

Per la realizzazione del sistema si è scelto di usare un'architettura di tipo Repository, che garantisce come sappiamo un disaccoppiamento in sottosistemi capaci di comunicare unicamente con il sottosistema del Database ma non tra loro. Il vantaggio di tale architettura è quella di avere un approccio modulare per favorirne scalabilità, facilità di implementazione e gestione di eventuali problemi. Ogni sottosistema è dotato di un'interfaccia con un controllore sottostante, che contiene la logica delle funzioni interessate e gestisce le richieste da fare al sottosistema DBMSBoundary, il quale si occupa proprio di comunicare con il nodo del DBMS. Interfacce, controllori e DBMSBoundary di ogni sottosistema risiedono sul proprio nodo ma il DBMS in un nodo assestante.

Per la realizzazione di questo software è stato scelto Apache NetBeans come ambiente di sviluppo, sia per la comodità d'implementazione delle librerie Java e l'interfaccia grafica sia per la creazione del design, che sono proprie della libreria Java Swing.

Per alleggerire l'implementazione di tutto il sistema, è stato scelto di usufruire di Ngrok, proxy inverso distribuito a livello globale che gestisce i servizi Web in esecuzione su qualsiasi cloud o rete privata. Per l'hosting del software è stato utilizzato XAMPP, multiplatforma software libera costituita da Apache HTTP Server, il DBMS MariaDB.

Per interfacciarsi al database, si ha usufruito del DBMS MariaDB fornita da XAMPP attraverso l'interfaccia PHPMyAdmin: applicazione web scritta in PHP, distribuita con licenza GPL che consente di amministrare un database MariaDB tramite qualsiasi browser.

Per quanto riguarda l'invio dell'e-mail nel caso di recupero password, si è scelto di usare un indirizzo di posta elettronica fornito da Google chiamato: smtp.gmail.com. Una volta settate le proprietà di questo host, viene inizializzata una sessione in cui l'e-mail mittente a nostra scelta effettua l'autenticazione in background, successivamente invia l'e-mail al destinatario, ovvero quella inserita nel campo e-mail durante il caso d'uso "Recupero Credenziali".

Packages

Per un'organizzazione ordinata e semplice durante lo sviluppo del software, si è scelto di suddividere le componenti del sistema in diversi package. A seguire, l'elenco dei package:

Package	Descrizione
GestioneFarmaci	<p>Contiene i pacchetti da installare sul sistema del Farmacista. Le interfacce sono state realizzate mediante JavaSwing, e le control di sua appartenenza vengono utilizzate per svolgere diverse mansioni:</p> <ul style="list-style-type: none">• Ordinare farmaci.• Modificare un ordine.• Modificare i parametri di un contratto.• Conferma la ricezione degli ordini, con carico merci.• Tracciabilità degli ordini.
GestioneAzienda	<p>Contiene i pacchetti da installare sul sistema del Magazziniere. Le interfacce sono state realizzate mediante JavaSwing, e le control di sua appartenenza vengono utilizzate per svolgere la seguente mansione:</p> <ul style="list-style-type: none">• Supervisionare e risolvere ordini con stato di errore.
GestioneCorriere	<p>Contiene i pacchetti da installare sul sistema del Fattorino. Le interfacce sono state realizzate mediante JavaSwing, e le control di sua appartenenza vengono utilizzate per svolgere le seguenti mansioni:</p> <ul style="list-style-type: none">• Visionare gli ordini assegnati automaticamente dal sistema.• Consegnare gli ordini al Farmacista attraverso una firma digitale inserita nel suo palmare, assegnatogli dall'azienda farmaceutica.
Autenticazione	<p>Contiene gli elementi necessari per far sì che l'utente possa effettuare l'autenticazione al sistema. Le interfacce sono state realizzate mediante JavaSwing, e le control di sua appartenenza vengono utilizzare per diverse mansioni:</p> <ul style="list-style-type: none">• Verifica delle credenziali inserite da parte di un utente.• Inviare un e-mail in caso di recupero password da parte di un utente.
Utils	<p>Contiene classi wrapper per API. Il pacchetto è specializzato per il sistema, e quindi poco riutilizzabile in altri contesi.</p>
Entity	<p>Contiene le classi che modellano le entità di cui è necessaria la rappresentazione nel sistema.</p>
Common	<p>Contiene classi comuni ai sottosistemi Farmacia, Corriere e Azienda, ad esempio la classe <i>CredenzialiUtente</i>.</p>

org.mariadb.jdbc

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

JavaSwing

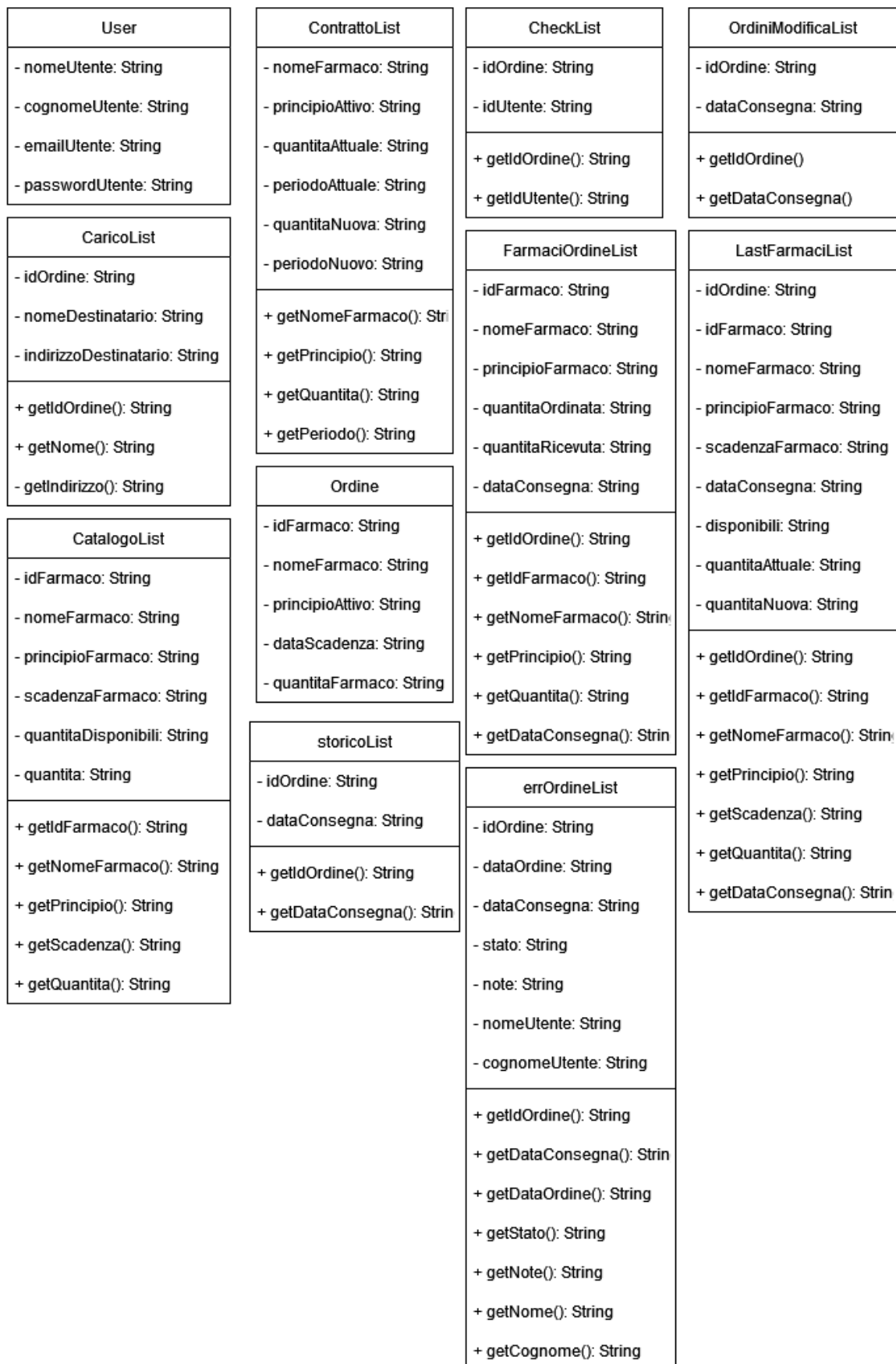
Il package è utilizzato per la realizzazione e la gestione delle interfacce grafiche dell'utente.

Java

Contiene i package e le librerie standard di Java

- **SQL:**
 - È un package utilizzato per gestire le comunicazioni con il DBMS, in particolare permette di eseguire tutte le query necessarie per il nostro sistema.
- **Mail:**
 - È un package utilizzato per gestire la comunicazione tra utente ed host gmail, in particolare viene utilizzato per inviare un'e-mail contenente la password da recuperare.
- **Utils:**
 - Contiene le strutture dati utilizzate per la realizzazione del sistema.
- **Time:**
 - API per le date, il tempo, gli istanti e le durate.

Entity



Utils

Utils
- creaAreaUtente(w: int, h: int) + creaSchermataConferma(messaggio: String)

Mail
+ RecuperaCredenziali (msg: String, dest: String, subject: String)

«Boundary» DBMSBoundary
- connessione: Connection
+ queryVerifica (email: String, password: String): boolean + queryVerifica (email: String): boolean + queryOrdineDaConsegnare(): Ordine[*] + queryVerificaCredenziali (email: String, password: String): boolean + queryCambioStato (stato: String): String + queryCercaOrdini(): boolean + queryAssegnaOrdine(): OrdineAssegnato[*] + queryDiRichiesta(): OrdineAssegnato[*] + queryCatalogo(): Catalogo[*] + queryRegistraOrdine(): OrdineRegistrato[*] + queryVerificaOrdine(): boolean + queryCheckFarmaco(): String + queryCreaOrdine(): Ordine[*] + queryRimuoviFarmaciOrdine(): Ordine[*], Farmaci[*] + queryDataAttuale(): LocalDate + queryConfrontoDate(dataAttuale: LocalDate, dataScadenza: LocalDate): boolean + queryRimuoviFarmaciScadenza(): Farmaci[*] + queryAggiungiScorte(): Farmaci[*] + queryRimuoviFarmaciScaduti(): Farmaci[*] + queryContratto(idUtente: String, idContratto: String): Contratto[*] + queryModifica(idFarmaco: String, quantità: String, periodicità: String): Contratto[*] + queryPeriodicità(periodicità: String): String + queryCercaOrdine(idOrdine: String): Ordine[*] + queryChiediDettagli(idOrdine): boolean + querySegnalaErrore(stato: String, note: String): String + queryControlloQuantità(quantità: String): boolean + queryStoricoOrdine(idOrdine: String, idUtente: String): Ordini[*] + queryOrdiniErrati(idOrdine: String, idUtente: String): OrdiniErrati[*] + queryCorreggiStato(idOrdine: String, idUtente: String, stato: String, note: String, ordineCaricato: boolean) + queryTracciamento(idOrdine: String, idUtente: String): Ordini[*] + connect() + connectDBMS()

Common

«Boundary» SchermataConferma
- bottoneConferma: Button
+ conferma()

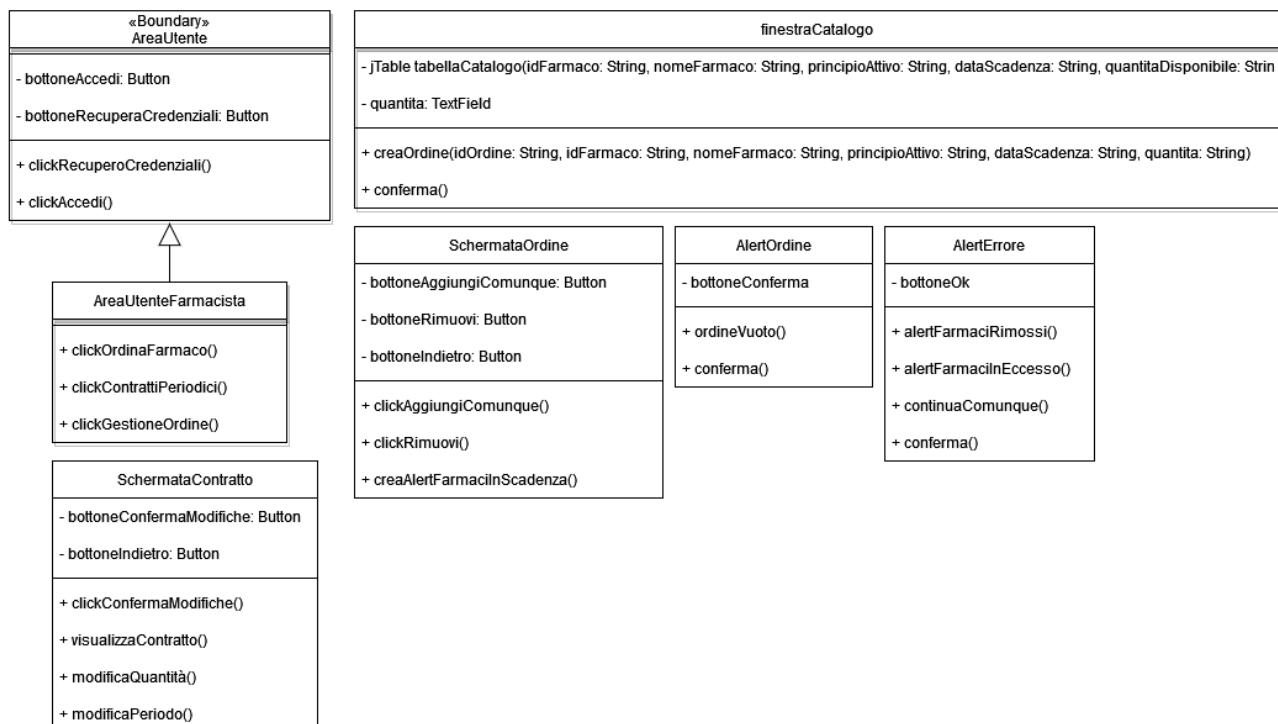
«Boundary» AreaUtente
- bottoneAccedi: Button
- bottoneRecuperaCredenziali: Button
+ clickRecuperoCredenziali()
+ clickAccedi()

GestioneFarmaci

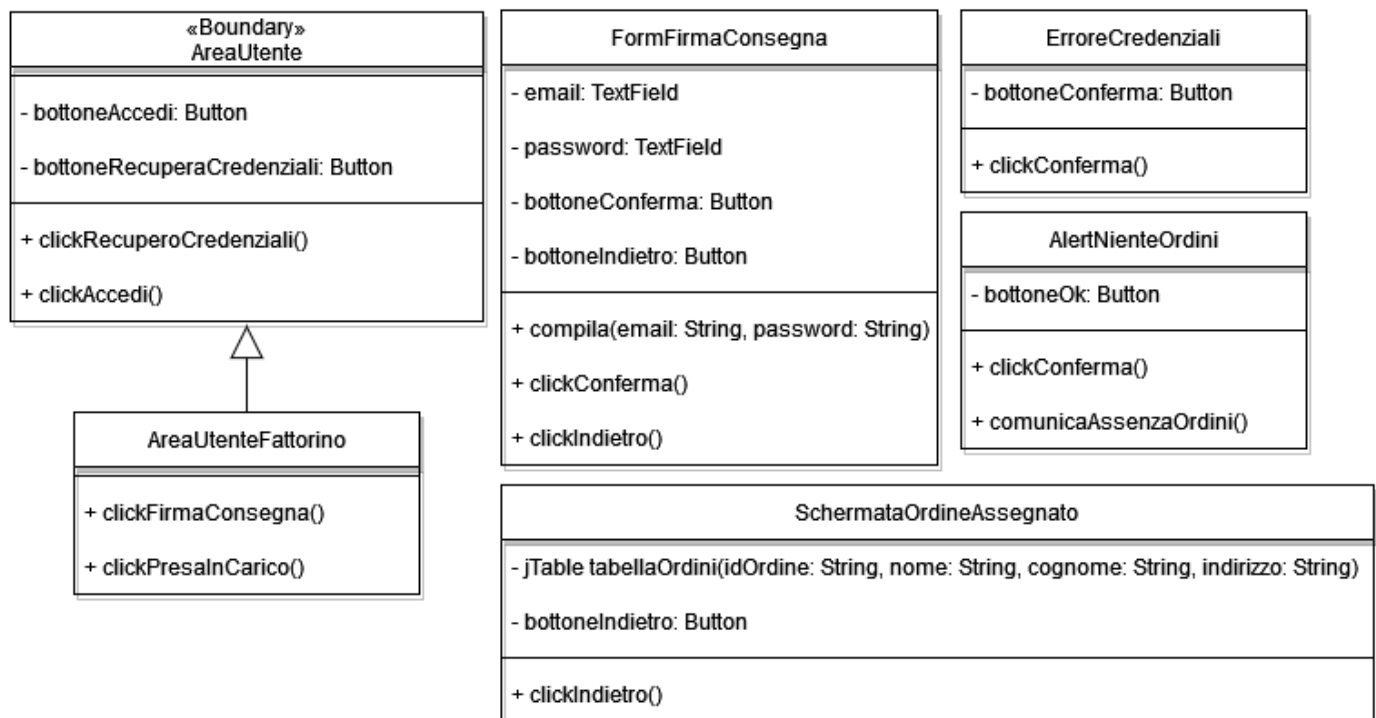
GestioneFarmaci.Control

GestoreCatalogo	GestoreControllo	GestoreScorte	GestoreContratto
+ creaCatalogoList() + creaFinestraCatalogo()	+ farmaciNonDisponibili() + farmaciInScadenza() + aggiungiComunque() + rimuovi() + creaAlertErrore() + creaSchermataOrdine() + chiediDataAttuale()	+ chiediDataAttuale() + periodicitàScorte() + confrontoDate(dataAttuale, periodo) + scadenzaScorte() + controlloScadenza()	+ creaSchermataContratto() + creaContrattoList + creaSchermataConferma()
gestoreOrdine			GestoreOrdinePeriodico
+ isEmpty() + creaSchermataConferma() + creaAlertOrdine() + creaOrdine()			+ chiediDataAttuale() + confrontoDate(dataAttuale, periodo)

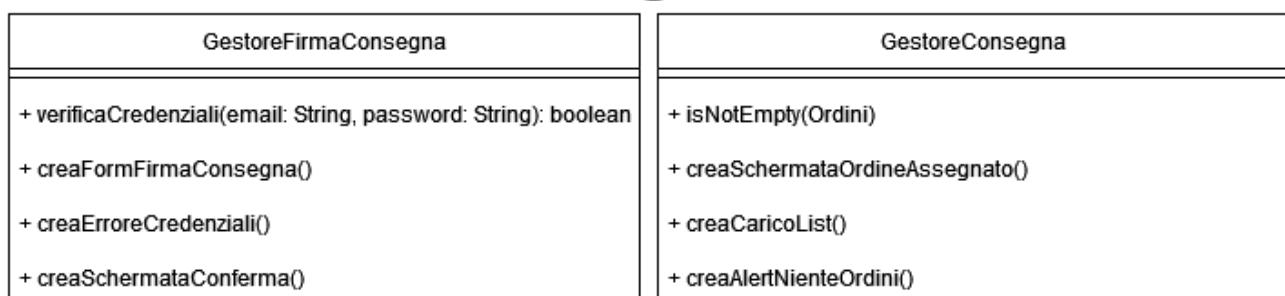
GestioneFarmaci.Interface



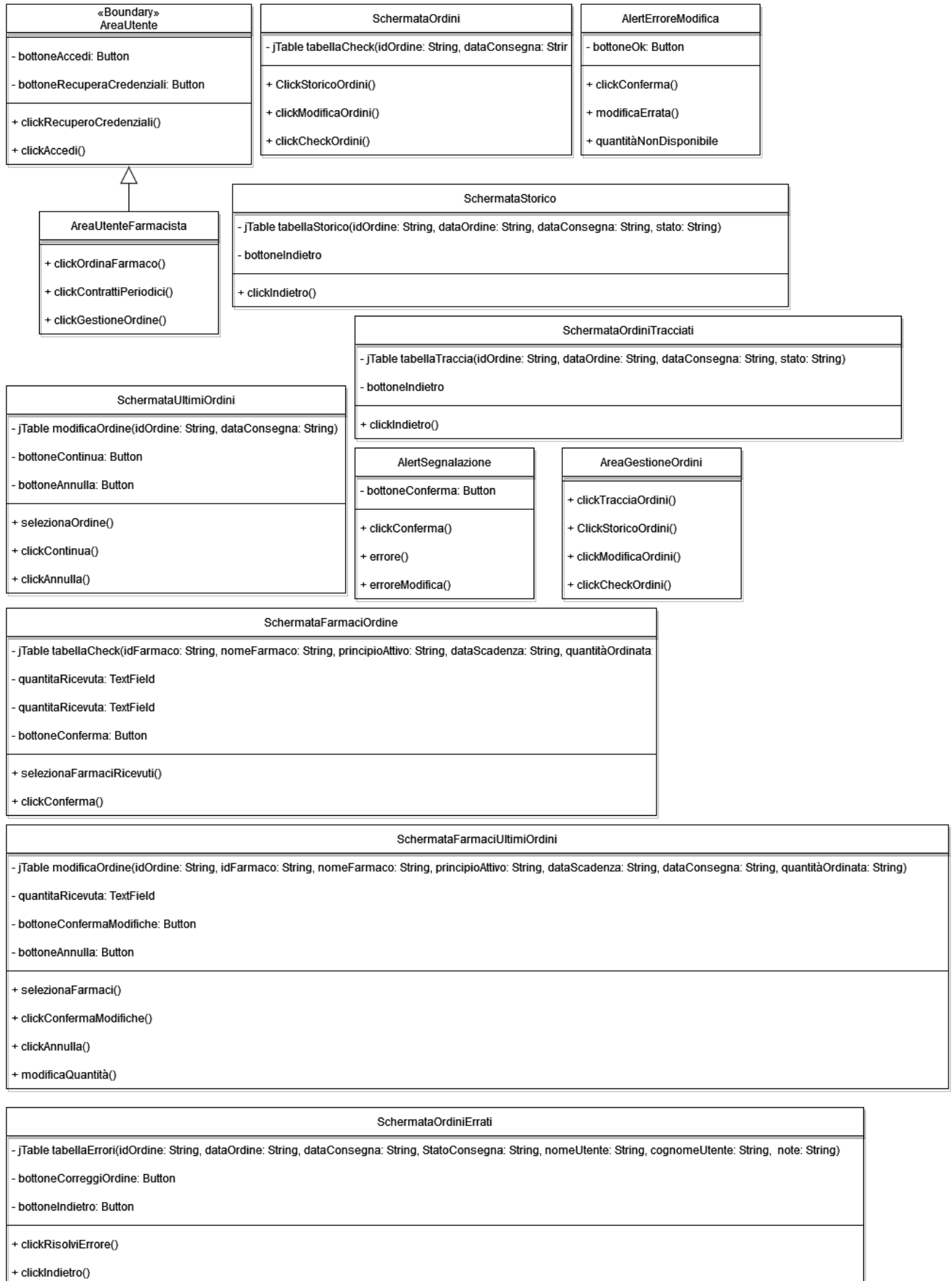
GestioneConsegne.Interface



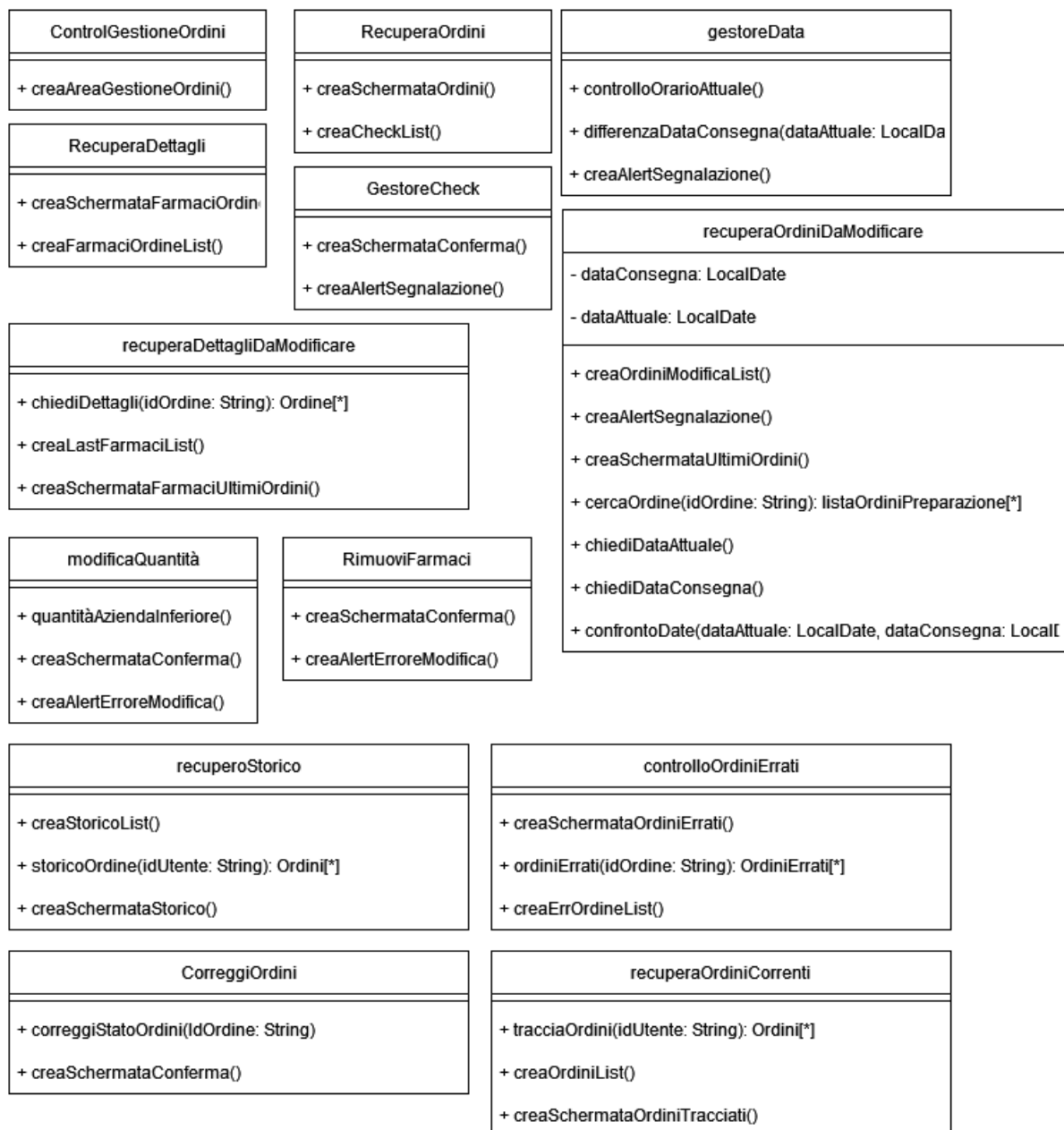
GestioneConsegne.Control



GestioneOrdine.Interface



GestioneOrdine.Control



Autenticazione.Control

gestoreLogin	gestoreRecupero
+ verificaUtente(email: String, password: String): boolean + creaUser() + creaAreaUtente(idUtente: String) + creaErroreAccesso() + creaModuloLogin()	+ creaSchermataRecupero() + verificaEmail(email: String): boolean + creaErroreRecupero() + creaSchermataConferma() + isEmpty()

Autenticazione.Interface

ErroreAccesso	ModuloLogin	SchermataRecupero
- bottoneConferma: Button	- email: TextField - password: TextField - bottoneAccedi: Button - bottoneRecupero: Button	- email: TextField - bottoneConferma: Button - bottoneIndietro: Button
+ clickConferma()	+ clickAccedi() + clickRecuperoCredenziali()	+ clickConferma() + inserisciEmail()