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	
Utils	6
Common	
GestioneFarmaci	
GestioneConsegne	
GestioneOrdine	
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, multipiattaforma 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:

- Deales as	Descriptions
Package	Descrizione
GestioneFarmaci	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: Ordinare farmaci. Modificare un ordine. Modificare i parametri di un contratto. Conferma la ricezione degli ordini, con carico merci. Tracciabilità degli ordini.
GestioneAzienda	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: • Supervisionare e risolvere ordini con stato di errore.
GestioneCorriere	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: • 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	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:
Utils	 Verifica delle credenziali inserite da parte di un utente. Inviare un e-mail in caso di recupero password da parte di un utente. Contiene classi wrapper per API. Il pacchetto è specializzato per il sistema, e quindi poco riutilizzabile in altri contesi.
Entity	Contiene le classi che modellano le entità di cui è necessaria la rappresentazione nel sistema.
Common	Contiene classi comuni ai sottosistemi Farmacia, Corriere e Azienda, ad esempio la classe <i>CredenzialiUtente</i> .

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:

o API per le date, il tempo, gli istanti e le durate.

Object Design UML Entity

Entity

- nomeUtente: String
- cognomeUtente: String
- emailUtente: String
- passwordUtente: String

CaricoList

- idOrdine: String
- nomeDestinatario: String
- indirizzoDestinatario: String
- + getIdOrdine(): String
- + getNome(): String
- getIndirizzo(): String

CatalogoList

- idFarmaco: String
- nomeFarmaco: String
- principioFarmaco: String
- scadenzaFarmaco: String
- quantitaDisponibili: String
- quantita: String
- + getIdFarmaco(): String
- + getNomeFarmaco(): String
- + getPrincipio(): String
- + getScadenza(): String
- + getQuantita(): String

ContrattoList

- nomeFarmaco: String
- principioAttivo: String
- quantitaAttuale: String
- periodoAttuale: String
- quantitaNuova: String
- periodoNuovo: String
- + getNomeFarmaco(): Stri
- + getPrincipio(): String
- + getQuantita(): String
- + getPeriodo(): String

Ordine

- idFarmaco: String
- nomeFarmaco: String
- principioAttivo: String
- dataScadenza: String
- quantitaFarmaco: String

storicoList

- idOrdine: String
- dataConsegna: String
- + getIdOrdine(): String
- + getDataConsegna(): Strin

CheckList

- idOrdine: String
- idUtente: String
- + getIdOrdine(): String
- + getIdUtente(): String

FarmaciOrdineList

- idFarmaco: String
- nomeFarmaco: String
- principioFarmaco: String
- quantitaOrdinata: String
- quantitaRicevuta: String
- dataConsegna: String
- + getIdOrdine(): String
- + getIdFarmaco(): String
- getNomeFarmaco(): String
- + getPrincipio(): String
- + getQuantita(): String
- + getDataConsegna(): Strin

errOrdineList

- idOrdine: String
- dataOrdine: String
- dataConsegna: String
- stato: String
- note: String
- nomeUtente: String
- cognomeUtente: String
- + getIdOrdine(): String
- + getDataConsegna(): Strin
- + getDataOrdine(): String
- + getStato(): String
- + getNote(): String
- + getNome(): String
- + getCognome(): String

OrdiniModificaList

- idOrdine: String
- dataConsegna: String
- + getIdOrdine()
- getDataConsegna()

LastFarmaciList

- idOrdine: String
- · idFarmaco: String
- nomeFarmaco: String
- principioFarmaco: String
- scadenzaFarmaco: String
- dataConsegna: String
- disponibili: String
- quantitaAttuale: String
- quantitaNuova: String
- + getIdOrdine(): String
- + getIdFarmaco(): String
- + getNomeFarmaco(): String
- + getPrincipio(): String
- + getScadenza(): String
- + getQuantita(): String
- getDataConsegna(): Strin

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: LocalData, 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(quantita: 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

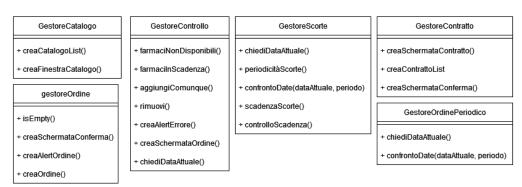
Common

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

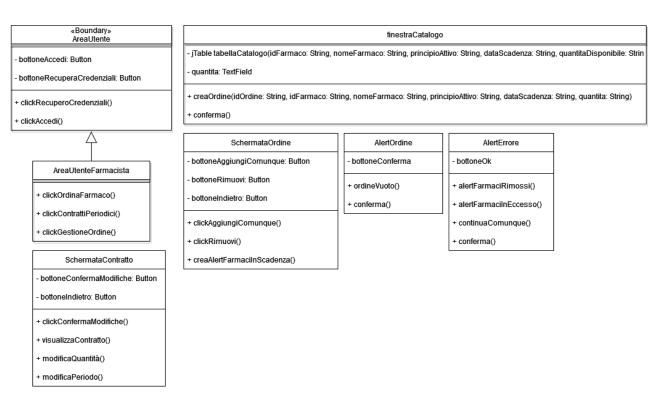
«Boundary»	
AreaUtente	
- bottoneAccedi: Button	
- bottoneRecuperaCredenziali: Button	
+ clickRecuperoCredenziali()	
+ clickAccedi()	

GestioneFarmaci

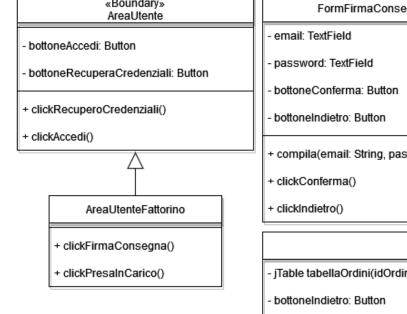
GestioneFarmaci.Control



GestioneFarmaci.Interface



GestioneConsegne.Interface



FormFirmaConsegna	ErroreCredenziali
mail: TextField	- bottoneConferma: Button
assword: TextField	+ clickConferma()
ottoneConferma: Button	
ottoneIndietro: Button	AlertNienteOrdini
	- bottoneOk: Button
compila(email: String, password: String)	Donotto Danott
zompiia(omaii. oamg, password. oamg)	

+ clickindletro()		
SchermataOrdineAssegnato		
- jTable tabellaOrdini(idOrdine: String, nome: String, cognome: String, indirizzo: String)		
- bottoneIndietro: Button		
+ clickIndietro()		

+ clickConferma()

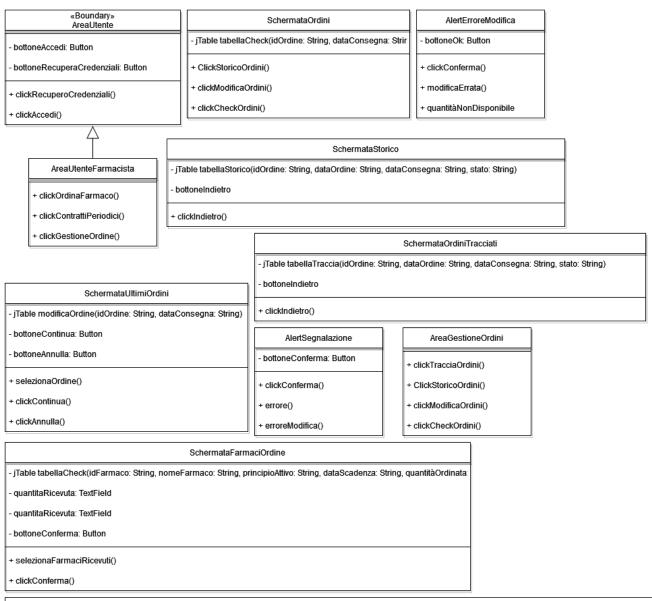
comunicaAssenzaOrdini()

GestioneConsegne.Control

GestoreFirmaConsegna	GestoreConsegna
+ verificaCredenziali(email: String, password: String): boolean	+ isNotEmpty(Ordini)
+ creaFormFirmaConsegna()	+ creaSchermataOrdineAssegnato()
+ creaErroreCredenziali()	+ creaCaricoList()
+ creaSchermataConferma()	+ creaAlertNienteOrdini()

GestioneOrdine

GestioneOrdine.Interface



SchermataFarmaciUltimiOrdini - jTable modificaOrdine(idOrdine: String, idFarmaco: String, nomeFarmaco: String, principioAttivo: String, dataScadenza: String, dataConsegna: String, quantitàOrdinata: String) - quantitaRicevuta: TextField - bottoneConfermaModifiche: Button - bottoneAnnulla: Button + selezionaFarmaci() + clickConfermaModifiche() + clickAnnulla() + modificaQuantità()

SchermataOrdiniErrati	
- jTable tabellaErrori(idOrdine: String, dataOrdine: String, dataConsegna: String, StatoConsegna: String, nomeUtente: String, cognomeUtente: String, note: String)	
- bottoneCorreggiOrdine: Button	
- bottoneIndietro: Button	
+ clickRisolviErrore()	
+ clickIndietro()	

GestioneOrdine.Control

ControlGestioneOrdini	RecuperaOrdin	ini gestoreData
	rtosaporasiani	
+ creaAreaGestioneOrdini()	+ creaSchermataOrdini	ni() + controlloOrarioAttuale()
RecuperaDettagli	+ creaCheckList()	+ differenzaDataConsegna(dataAttuale: LocalDa
+ creaSchermataFarmaciOrdin	GestoreCheck	+ creaAlertSegnalazione()
+ creaFarmaciOrdineList()	+ creaSchermataConfe	ferma() recuperaOrdiniDaModificare
	+ creaAlertSegnalazion	- dataConsegna: LocalDate
	orou noncognanazion	- dataAttuale: LocalDate
recuperaDettagliDa	Modificare	
+ chiediDettagli(idOrdine: String)	: Ordine[*]	+ creaOrdiniModificaList() + creaAlertSegnalazione()
+ creaLastFarmaciList()		area Cohermata I Himi Ordini /
+ creaSchermataFarmaciUltimiO	rdini()	+ creaSchermataUltimiOrdini() + cercaOrdine(idOrdine: String): listaOrdiniPreparazione[*]
modificaQuantità	RimuoviFarmaci	+ chiediDataAttuale()
+ quantitàAziendaInferiore() + creaSchermataConferma()	+ creaSchermataConferr + creaAlertErroreModifica	+ confrontoDate(dataAttuale: LocalDate, dataConsegna: Local
+ creaAlertErroreModifica()		
recuperoSto	rico	controlloOrdiniErrati

recuperoStorico	controlloOrdiniErrati
+ creaStoricoList()	+ creaSchermataOrdiniErrati()
+ storicoOrdine(idUtente: String): Ordini[*]	+ ordiniErrati(idOrdine: String): OrdiniErrati[*]
+ creaSchermataStorico()	+ creaErrOrdineList()
CorreggiOrdini	recuperaOrdiniCorrenti
- Control of the cont	
+ correggiStatoOrdini(ldOrdine: String)	+ tracciaOrdini(idUtente: String): Ordini[*]
+ creaSchermataConferma()	+ creaOrdiniList()
	+ creaSchermataOrdiniTracciati()

Autenticazione

Autenticazione.Control

gestoreLogin + verificaUtente(email: String, password: String): boolean + creaUser() + creaAreaUtente(idUtente: String) + creaErroreAccesso() + creaModuloLogin() + isE

gestoreRecupero
+ creaSchermataRecupero()
+ verificaEmail(email: String): boolean
+ creaErroreRecupero()
+ creaSchermataConferma()
+ isEmpty()

Autenticazione.Interface

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