

FarmaByte

Lorenzo Guerra 00880376

Guglielmo Palaferri 00873714

Giacomo Romanini 00874849

Luglio 2021

Indice

1	Abstract	1
2	Documento dei Requisiti	2
2.1	Raccolta dei requisiti	2
2.2	Tabella dei Requisiti	3
2.3	Analisi dei Requisiti	4
2.3.1	Vocabolario	4
2.3.2	Casi d'uso della farmacia	5
2.3.3	Scenari	6
2.4	Analisi del Rischio	13
2.4.1	Tabella Valutazione dei Beni	13
2.4.2	Tabella Minacce/Controlli	14
2.4.3	Analisi Tecnologica della Sicurezza	14
2.4.4	Security Use Case & Misuse Case	15
2.4.5	Security Use Case & Misuse Case Scenari	16
2.4.6	Requisiti di Protezione dei Dati	17
3	Analisi del Problema	18
3.1	Analisi Documento dei Requisiti: Analisi delle Funzionalità	18
3.1.1	Analisi Documento dei Requisiti: Analisi dei Vincoli	22
3.1.2	Analisi Documento dei Requisiti: Analisi delle Interazioni	23
3.1.3	Analisi Ruoli e Responsabilità	24
3.1.4	Scomposizione del Problema	25
3.1.5	Creazione Modello del Dominio	27
3.1.6	Architettura Logica: Struttura	28
3.1.7	Architettura Logica: Interazione	31
3.1.8	Architettura Logica: Comportamento	34
3.1.9	Piano di Lavoro	35
3.1.10	Piano del Collaudo	36
4	Progettazione	39
4.1	Progettazione Architetturale	39
4.1.1	Requisiti non funzionali	39
4.1.2	Scelte tecnologiche	39
4.1.3	Scelta dell'architettura	39
4.1.4	Pattern architetturali e di design	40
4.2	Progettazione di dettaglio	43
4.2.1	Struttura	43
4.2.2	Interazione	53
4.3	Progettazione della persistenza	59
4.3.1	Formato dei file di log	60
4.4	Progettazione del collaudo	61
4.5	Progettazione per il deployment	62
4.6	Deployment	63
4.6.1	Artefatti	63
4.6.2	Deployment Type-Level	63

1 Abstract

Il progetto riguarda la creazione di un applicativo software per la gestione interna dell'inventario delle farmacie e per l'interfacciamento con i clienti. Il cliente ha la possibilità di cercare quale sia la farmacia più vicina ad avere un certo medicinale e, dopo essersi autenticato, può inviare una prenotazione del farmaco. La farmacia quindi può ricevere la prenotazione del cliente registrato al servizio che verrà poi finalizzata in acquisto. Inoltre la gestione dell'inventario del negozio permette alle farmacie di controllare l'elenco dei farmaci disponibili e ne facilita la gestione degli ordini e delle rimanenze, segnalando all'operatore le medicine in scadenza e in esaurimento. L'applicativo fornisce anche una funzionalità di ricerca farmaci, per i clienti che desiderano verificare la disponibilità di un farmaco in una certa località.

2 Documento dei Requisiti

2.1 Raccolta dei requisiti

- I clienti delle farmacie hanno a disposizione due servizi: controllare se un farmaco è disponibile vicino alla loro posizione e/o prenotarlo.
- Il cliente fornisce la sua posizione che l'applicativo userà per indicargli le farmacie più vicine. Contemporaneamente specificherà il farmaco da cercare e l'applicativo fornirà le 10 farmacie più vicine ad averlo in magazzino indicandone se è disponibile o sta per terminare.
- Per la prenotazione è necessario possedere un account
- La prenotazione sarà composta da uno o più farmaci, dalla farmacia, e dal giorno.
- L'account viene creato in due fasi: 1. Registrazione con nome, cognome, password, data di nascita, email e codice fiscale 2. Autenticazione di persona in farmacia
- La email deve essere univoca, la password di almeno 8 caratteri, contenente almeno un numero e un carattere alfabetico.
- Per l'autenticazione è necessario mostrare il tesserino sanitario per l'identificazione in farmacia.
- Il cliente può vedere la lista delle sue prenotazioni in corso
- Il farmacista vede le prenotazioni, i farmaci disponibili in negozio e viene segnalato riguardo ai farmaci in esaurimento
- Il farmacista può confermare le prenotazioni andate a buon fine
- Se alla fine della giornata un utente non si presenta allora l'evento viene registrato, per poi avvisare il farmacista che può eventualmente bloccare l'utente per 1 mese.
- Il sistema sarà ovviamente distribuito e di natura client-server con la presenza di un database centrale dove memorizzare i dati
- La gestione delle vendite, degli ordini e modifiche di magazzino è gestita da un altro software
- Non va considerata la gestione dei dati del personale

2.2 Tabella dei Requisiti

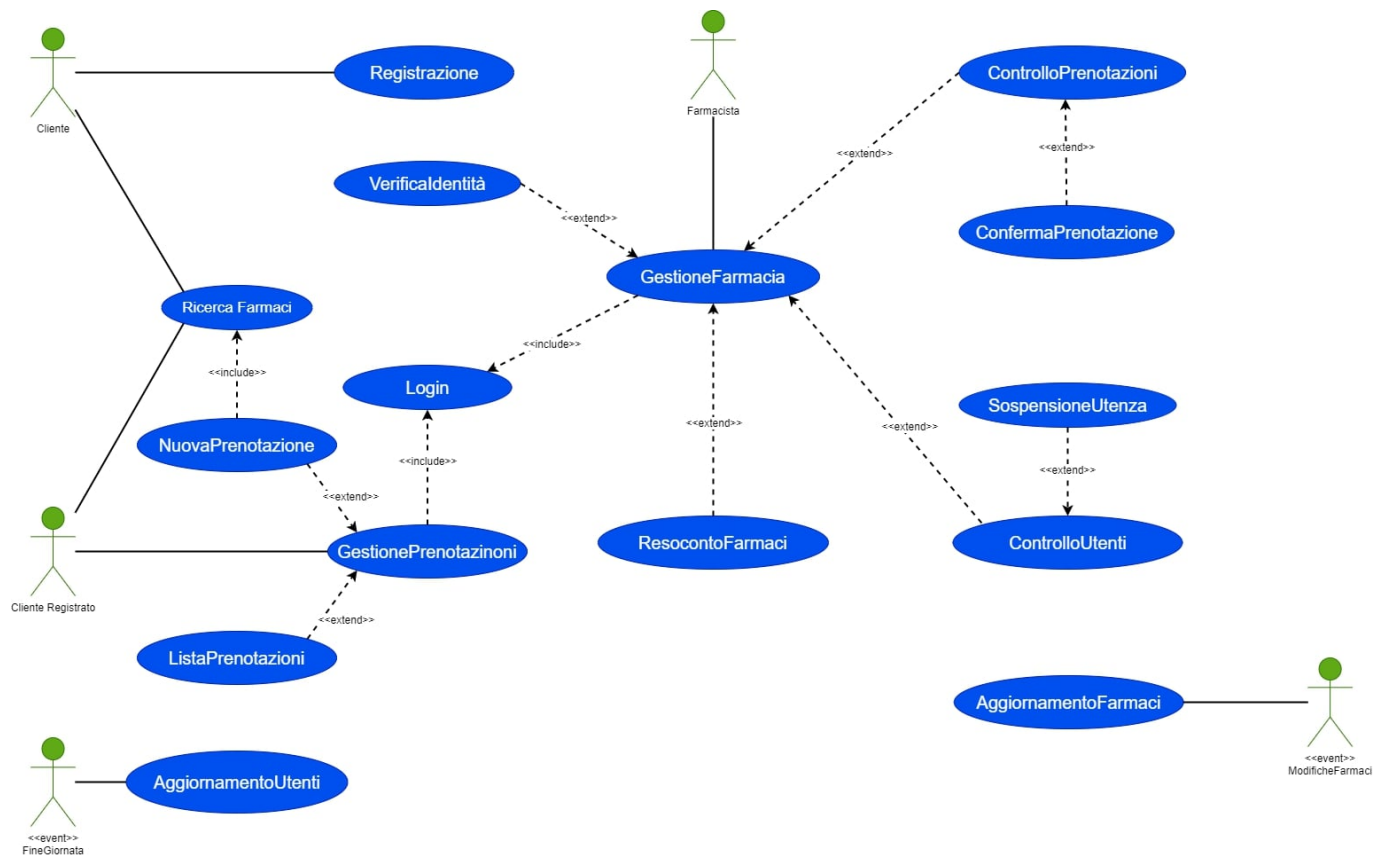
ID	Requisiti	Tipo
R1F	Localizzazione delle farmacie più vicine in base al farmaco da cercare	Funzionale
R2F	Specifica del farmaco da cercare da parte dell'utente	Funzionale
R3F	Presentazione delle farmacie che dispongono di un farmaco	Funzionale
R4F	Registrazione di un account tramite l'interfaccia web	Funzionale
R5F	Attivazione dell'account con identificazione fisica dell'utente con documento	Funzionale
R6F	La prenotazione sarà composta da uno o più farmaci, dalla farmacia e dal giorno	Funzionale
R7F	Identificazione attraverso email univoca e password di almeno 8 caratteri, contenente almeno un carattere alfabetico e un carattere numerico	Funzionale
R8F	Visualizzazione delle prenotazioni del cliente	Funzionale
R9F	Visualizzazione delle prenotazioni della farmacia	Funzionale
R10F	Visualizzazione del numero dei farmaci disponibili	Funzionale
R11F	Notifica dei farmaci in esaurimento o in scadenza	Funzionale
R12F	Conferma della prenotazione andata a buon fine	Funzionale
R13F	Notifica della mancata finalizzazione in acquisto di una prenotazione	Funzionale
R14F	Blocco dell'utente che effettua troppe prenotazioni senza presentarsi	Funzionale
R15F	Verrà memorizzato il numero di prenotazioni andate a buon fine	Funzionale
R1NF	Velocità di memorizzazione dei dati	Non Funzionale
R2NF	Velocità della ricerca dei dati	Non Funzionale
R3NF	Semplicità dell'interfaccia	Non Funzionale
R4NF	Un utente non può avere più di un account verificato	Non Funzionale
R5NF	la gestione delle vendite e ordini è gestita da un altro software	Non Funzionale
R6NF	Per prenotare l'utente deve essere registrato	Non Funzionale
R7NF	La gestione delle vendite, degli ordini e modifiche di magazzino è gestita da un altro software	Non Funzionale
R8NF	Non va considerata la gestione dei dati del personale	Non Funzionale

2.3 Analisi dei Requisiti

2.3.1 Vocabolario

Voce	Definizione	Sinonimi
Cliente	Persona che usufruisce del servizio lato cliente	Utente
ClienteRegistrato	Cliente che possiede un account identificato con cui può effettuare prenotazioni	UtenteRegistrato
Farmacia	Farmacia che aderisce al servizio	Punto vendita
Farmaco	Medicinale che viene venduto in farmacia	
Farmacista	Utente che accede con le credenziali della farmacia	Operatore
Prenotazione	Richiesta di farmaci da comprare in negozio	
Data e ora prenotazione	Indicazione temporale del momento in cui avverrà la prenotazione	
Posizione	Luogo della ricerca o collocamento geografico della farmacia	
Credenziali	Insieme composto da email e password necessari per accedere al sistema	
Email	Indirizzo di posta elettronica del cliente utilizzata anche per l'autenticazione	
Password	Codice alfanumerico di almeno 8 caratteri	
Magazzino	Luogo fisico in cui vengono conservati i farmaci di un punto vendita	Deposito

2.3.2 Casi d'uso della farmacia



2.3.3 Scenari

Titolo	GestioneFarmacia
Descrizione	Gestione dell'utenza di un cliente registrato
Attori	Farmacista
Relazioni	Login, SospensioneUtenza, VerificaIdentità, ResocontoFarmaci, ResocontoPrenotazioni
Precondizioni	
Postcondizioni	
Scenario Principale	1. Login 2. Il farmacista può eseguire la verifica, sospendere un'account, controllare le prenotazioni e i farmaci in magazzino
Scenari Alternativi	
Requisiti non funzionali	Velocità di ricerca dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	ResocontoUtenti
Descrizione	Si controllano gli utenti con potenzialmente sospendibili
Attori	Farmacista
Relazioni	SospensioneUtenza, GestioneFarmacia
Precondizioni	
Postcondizioni	Viene mostrato l'elenco degli Utenti a rischio sospensione
Scenario Principale	1. Il Farmacista va nella schermata di visualizzazione utenti 2. Il sistema recupera l'elenco degli utenti a rischio 3. Il sistema mostra a video l'elenco degli utenti
Scenari Alternativi	
Requisiti non funzionali	Velocità di ricerca dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	ResocontoFarmaci
Descrizione	Viene mostrato l'elenco dei farmaci in scadenza o in esaurimento
Attori	Farmacista
Relazioni	GestioneFarmacia
Precondizioni	
Postcondizioni	Viene mostrato l'elenco degli Utenti a rischio sospensione
Scenario Principale	1. Il Farmacista va nella schermata di visualizzazione farmaci 2. Il sistema recupera l'elenco dei farmaci in esaurimento o in scadenza 3. Il sistema mostra a video l'elenco richiesto
Scenari Alternativi	
Requisiti non funzionali	Velocità di ricerca dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	ControlloPrenotazioni
Descrizione	Si controllano le prenotazioni non terminate
Attori	Farmacista
Relazioni	ConfermaPrenotazione,GestioneFarmacia
Precondizioni	
Postcondizioni	Viene mostrato l'elenco delle prenotazioni
Scenario principale	1. Il Farmacista va nella schermata di visualizzazione prenotazioni 2. Il sistema recupera l'elenco delle prenotazioni giornaliere 3. Il sistema mostra a video l'elenco delle prenotazioni
Scenari Alternativi	
Requisiti non funzionali	Velocità di ricerca dei dati
Punti aperti	

Titolo	ConfermaPrenotazione
Descrizione	Il Farmacista conferma la prenotazione avvenuta
Attori	Farmacista
Relazioni	ControlloPrenotazioni
Precondizioni	
Postcondizioni	La prenotazione viene confermata
Scenario principale	1. ControlloPrenotazioni 2. Il Farmacista conferma l'avvenuta prenotazione
Scenari Alternativi	
Requisiti non funzionali	Semplicità dell'interfaccia
Punti aperti	

Titolo	Registrazione
Descrizione	Il cliente si registra al servizio
Attori	Cliente
Relazioni	
Precondizioni	Il cliente dispone di un codice fiscale valido
Postcondizioni	Il cliente è registrato nel sistema ed è posto in attesa della verifica
Scenario principale	1. Il cliente accede alla sezione di registrazione 2. Il cliente inserisce i propri dati: nome, cognome, data di nascita, email, password e il codice fiscale 3. Il cliente termina la registrazione, se avvenuta con successo gli viene mostrata la conferma e viene reindirizzato alla pagina principale
Scenari Alternativi	Scenario a: il codice fiscale è già registrato 3. Il sistema verifica che è già presente un utente con quel codice fiscale, quindi notifica il cliente con un messaggio di errore.
Requisiti non funzionali	Semplicità dell'interfaccia
Punti aperti	

Titolo	VerificaIdentità
Descrizione	Verifica dell'identità dell'utente registrato
Attori	Cliente, Farmacista
Relazioni	GestioneFarmacia
Precondizioni	Il cliente è registrato
Postcondizioni	L'utente è stato verificato e il suo account viene abilitato per effettuare delle prenotazioni
Scenario principale	<ol style="list-style-type: none"> 1. Il cliente va in farmacia con il documento specificato in fase di registrazione 2. Il cliente viene identificato dal farmacista 3. Il farmacista chiede al sistema di recuperare l'utente 4. Il farmacista attiva l'account dell'utente
Scenari alternativi	4. Il sistema non trova nessun utente, segnala il farmacista
Requisiti non funzionali	Velocità di memorizzazione e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	SospensioneUtenza
Descrizione	Se un utente non ha concluso troppe prenotazioni allora viene proposta la sospensione dell'utente al farmacista
Attori	Farmacista
Relazioni	ResocontoPrenotazioni, GestioneFarmacia
Precondizioni	Il cliente è diffidato dal sistema (ha molte prenotazioni non concluse)
Postcondizioni	Il cliente non può più effettuare prenotazioni per 30 giorni
Scenario principale	<ol style="list-style-type: none"> 1. Si verifica l'evento FineGiornata 2. Il Sistema recupera l'elenco delle prenotazioni e lo analizza 3. Se il Sistema rileva un numero eccessivo di prenotazioni non concluse per un determinato utente, allora lo segnala al farmacista 4. Il farmacista, se ritiene necessario, può confermare la sospensione dell'utente
Scenari alternativi	
Requisiti non funzionali	Velocità nella ricerca dei dati e semplicità dell'interfaccia
Punti aperti	

Titolo	ResocontoPrenotazioni
Descrizione	Si controllano le prenotazioni non terminate per utente
Attori	Farmacista, FineGiornata
Relazioni	Resoconto, SospensioneUtenza, GestioneFarmacia
Precondizioni	
Postcondizioni	Viene mostrato l'elenco delle prenotazioni
Scenario principale	1. Il Farmacista va nella schermata di visualizzazione farmaci 2. Il sistema recupera l'elenco delle prenotazioni giornaliere 3. Il sistema mostra a video l'elenco delle prenotazioni
Scenari Alternativi	
Requisiti non funzionali	Velocità di ricerca dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	RicercaFarmaci
Descrizione	L'utente verifica la disponibilità di un particolare farmaco nelle farmacie più vicine a lui
Attori	Cliente
Relazioni	
Precondizioni	
Postcondizioni	Si visualizza la lista delle farmacie con disponibilità
Scenario principale	<ol style="list-style-type: none"> 1. Il cliente si reca nella pagina di ricerca 2. Il cliente inserisce il nome del farmaco per cui eseguire la ricerca 3. Il sistema ottiene la lista delle farmacie aventi il farmaco specificato entro un range dalla località specificata. 4. Il sistema ordina la lista in base alla distanza geografica dalla zona dell'utente, in ordine crescente 5. La lista viene mostrata all'utente
Scenari Alternativi	<p>Scenario alternativo A:</p> <ol style="list-style-type: none"> 1. Login 2. Il cliente si reca nella pagina di ricerca 3. Il cliente inserisce il nome del farmaco per cui eseguire la ricerca 4. Il sistema ottiene la lista delle farmacie aventi il farmaco specificato entro un range dalla località specificata. 5. Il sistema ordina la lista in base alla distanza geografica dalla zona dell'utente, in ordine crescente 6. La lista viene mostrata all'utente 7. L'utente può selezionare il farmaco per cominciare una prenotazione di quel farmaco nella farmacia scelta <p>Scenario alternativo B:</p> <ol style="list-style-type: none"> 1. Login 2. Il cliente avvia una nuova prenotazione dalla home 3. Il cliente seleziona la farmacia in cui effettuare la prenotazione 4. Il cliente inserisce il nome del farmaco per cui eseguire la ricerca 5. Il sistema ottiene la la lista dei farmaci disponibili nella farmacia scelta il cui nome inizia per il testo inserito dall'utente 6. La lista viene mostrata all'utente 7. L'utente può selezionare il farmaco corrispondente, se è presente nella lista, ed aggiungerlo alla prenotazione in corso
Requisiti non funzionali	Velocità di ricerca dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	NuovaPrenotazione
Descrizione	L'utente prenota a suo nome una lista di farmaci
Attori	ClienteRegistrato
Relazioni	GestionePrenotazioni
Precondizioni	
Postcondizioni	Il sistema ha memorizzato i dati della prenotazione, in attesa di conferma da parte della farmacia
Scenario principale	<ol style="list-style-type: none"> 1. Login 2. Il cliente seleziona i farmaci che vuole prenotare, la quantità, e inserisce la data di ritiro desiderata 3. Il cliente invia la richiesta di prenotazione 4. Il sistema pone la richiesta in attesa di conferma
Scenari Alternativi	<p>Scenario a: La farmacia non dispone dei farmaci richiesti.</p> <ol style="list-style-type: none"> 4. Il sistema nota che la farmacia non ha disponibilità di almeno uno dei farmaci specificati 5. Viene inviato al cliente un messaggio di errore
Requisiti non funzionali	Velocità di verifica dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	ListaPrenotazioni
Descrizione	L'utente ottiene la lista delle proprie prenotazioni passate ed in corso
Attori	ClienteRegistrato
Relazioni	GestionePrenotazioni
Precondizioni	
Postcondizioni	Al cliente viene mostrata la lista delle prenotazioni passate ed in corso
Scenario Principale	<ol style="list-style-type: none"> 1. Login 2. Il cliente seleziona l'opzione di visualizzazione della lista delle prenotazioni 3. Al cliente viene mostrato l'elenco delle prenotazioni effettuate
Scenari Alternativi	
Requisiti non funzionali	Velocità di verifica dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	GestionePrenotazioni
Descrizione	Gestione delle prenotazioni di un cliente registrato
Attori	ClienteRegistrato
Relazioni	Login, ListaPrenotazioni, NuovaPrenotazione
Precondizioni	
Postcondizioni	
Scenario Principale	1. Il cliente esegue il login 2. Il cliente può visualizzare le proprie prenotazioni passate o in corso e può effettuare nuove prenotazioni.
Scenari Alternativi	
Requisiti non funzionali	Velocità di verifica dei dati e semplicità di navigazione tra le diverse maschere
Punti aperti	

Titolo	Login
Descrizione	Permette di accedere al sistema
Attori	ClienteRegistrato, Farmacista
Relazioni	NuovaPrenotazione, GestioneFarmacia
Precondizioni	
Postcondizioni	L'utente ha accesso al sistema, limitato in base ai suoi privilegi
Scenario principale	1. L'utente inserisce le credenziali di accesso 2. Il sistema verifica le credenziali 3. Se le credenziali sono corrette, viene presentata la schermata iniziale
Scenari Alternativi	Scenario a: Credenziali non riconosciute. 3. Il sistema non riconosce le credenziali e rispedisce l'utente alla schermata di login con un messaggio di errore
Requisiti non funzionali	Velocità di verifica delle credenziali
Punti aperti	

Titolo	AggiornamentoUtenti
Descrizione	Aggiorna l'elenco degli utenti a rischio sospensione
Attori	FineGiornata
Relazioni	
Precondizioni	
Postcondizioni	Il DataBase degli utenti è aggiornato
Scenario principale	1. Si verifica l'evento FineGiornata 2. Il sistema controlla le prenotazioni non andate a buon fine 3. Il sistema aggiorna i dati relativi alle infrazioni degli utenti
Scenari Alternativi	
Requisiti non funzionali	Velocità della ricerca dei dati
Punti aperti	

Titolo	Aggiornamento Farmaci
Descrizione	Aggiorna l'elenco dei farmaci in magazzino
Attori	ModificheFarmaci
Relazioni	
Precondizioni	
Postcondizioni	Il DataBase dei farmaci è aggiornato
Scenario principale	1. Si verifica l'evento ModificheFarmaci 2. Il sistema recupera le modifiche dal DataBase Remoto 3. Il sistema aggiorna i dati relativi ai farmaci in magazzino
Scenari Alternativi	
Requisiti non funzionali	Velocità della ricerca dei dati
Punti aperti	

2.4 Analisi del Rischio

2.4.1 Tabella Valutazione dei Beni

Bene	Valore	Esposizione
Sistema Informativo	Alto. Fondamentale per il funzionamento del servizio	Alta. Perdita finanziaria e di immagine
Informazioni dei clienti	Alto. Dati generali dei clienti della farmacia, comprese le credenziali	Alta. Perdita di immagine dovuta alla divulgazione di dati sensibili
Informazioni relative al personale	Alto. Dati relativi ai farmacisti, incluse le credenziali di accesso all'area riservata	Molto Alta. Perdita finanziaria dovuta a usi impropri delle credenziali con privilegi elevati. Perdita di immagine possibile con la divulgazione dei dati relativi ai clienti
Dati delle prenotazioni	Alto. Necessario per tenere traccia delle prenotazioni	Molto Alta. Perdita finanziaria dovuta allo smarrimento di prenotazioni. Perdita di immagine con la divulgazione dei farmaci prenotati dai clienti

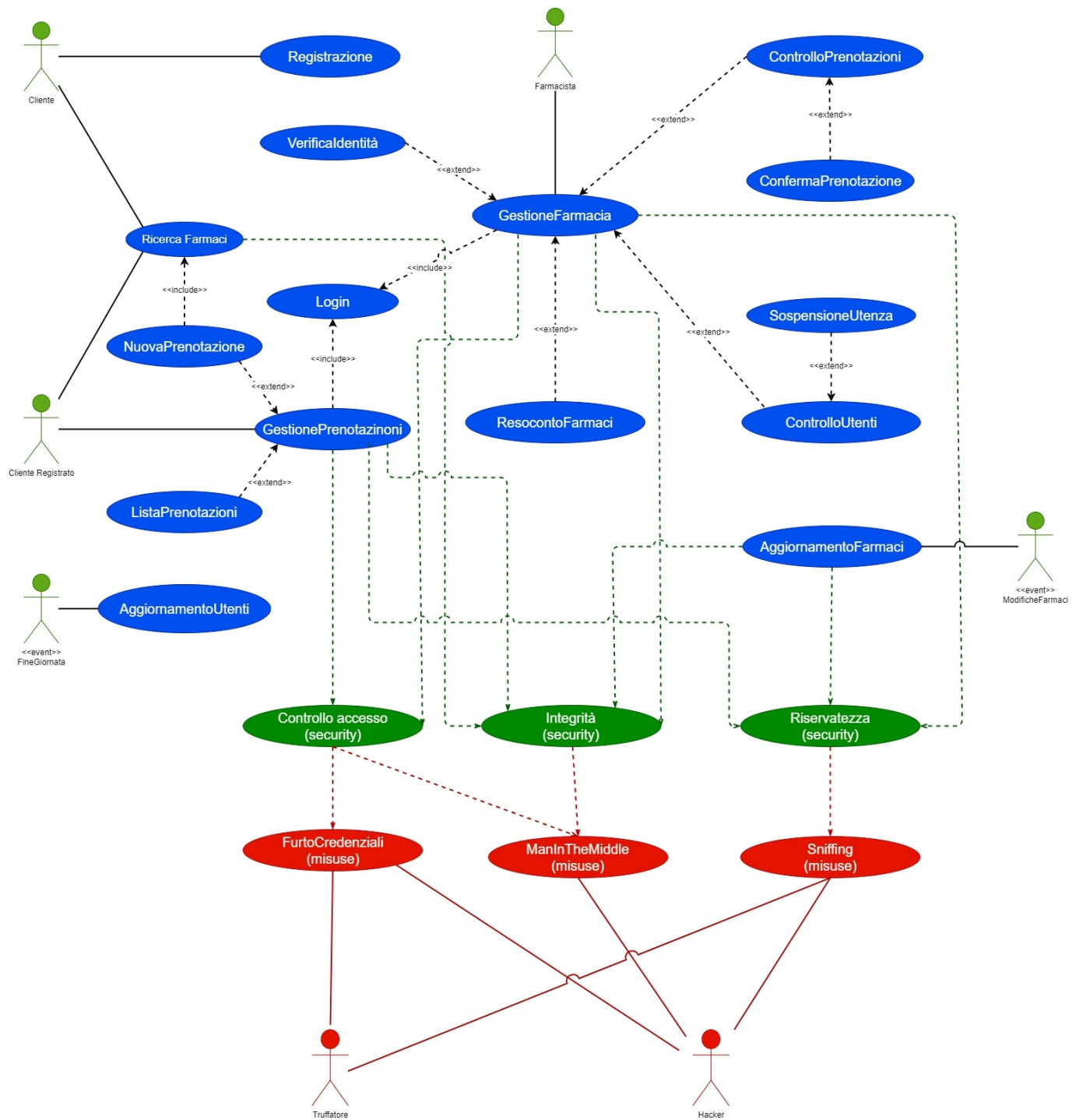
2.4.2 Tabella Minacce/Controlli

Minaccia	Probabilità	Controllo	Fattibilità
Furto credenziali Farmacista	Alta	Controllo sulla sicurezza della password - Log delle operazioni	Costo implementativo molto basso
Furto credenziali Cliente	Alta	Controllo sulla sicurezza della password - Log delle operazioni	Costo implementativo molto basso
Alterazione o intercettazione delle comunicazioni	Alta	Utilizzo di un canale sicuro - Log delle operazioni	Basso costo di realizzazione con determinati protocolli
Accesso non autorizzato al database	Bassa	Accesso da macchine sicure - Log di tutte le operazioni	Basso costo di realizzazione, il server deve essere ben custodito
DoS	Bassa	Controllo e limitazione delle richieste	Media complessità di implementazione
Saturazione del database	Bassa	1. Limitazione delle richieste in un dato intervallo di tempo. 2. Limite di tempo per la verifica di un cliente	Media complessità di implementazione

2.4.3 Analisi Tecnologica della Sicurezza

Tecnologia	Vulnerabilità
Autenticazione email/password	<ul style="list-style-type: none"> • Utente rivela volontariamente la password Utente rivela la password con un attacco di ingegneria sociale • Utente non esce dal sistema dopo aver eseguito le operazioni • Password banali
Cifratura comunicazioni	<ul style="list-style-type: none"> • In caso di cifratura simmetrica particolare attenzione va alla lunghezza delle chiavi ed alla loro memorizzazione • La memorizzazione è un fattore fondamentale anche nella cifratura asimmetrica
Architettura Client/Server	<ul style="list-style-type: none"> • DoS • Man in the Middle • Sniffing delle comunicazioni

2.4.4 Security Use Case & Misuse Case



2.4.5 Security Use Case & Misuse Case Scenari

Titolo	Riservatezza
Descrizione	I dati non sono accessibili da chi non ne ha i permessi
Misuse case	Sniffing
Relazioni	
Precondizioni	L'attaccante ha i mezzi per intercettare i messaggi del sistema
Postcondizioni	Il sistema impedisce all'attaccante di decifrare (in tempi utili) i messaggi intercettati
Scenario principale	<ol style="list-style-type: none"> 1. Il Sistema protegge i messaggi 2. L'attaccante riesce ad intercettare un messaggio 3. L'attaccante prova a decifrare i messaggi, ma non riesce a trovare un modo per farlo abbastanza velocemente
Scenari di un attacco avvenuto con successo	<ol style="list-style-type: none"> 1. Il Sistema protegge i messaggi 2. L'attaccante riesce ad intercettare un messaggio 3. L'attaccante riesce a decifrare i messaggi e a leggerne il contenuto, ma solamente per una sessione di un utente

Titolo	Integrità
Descrizione	Integrità dei dati del sistema
Misuse case	ManInTheMiddle
Relazioni	
Precondizioni	<ol style="list-style-type: none"> 1. L'attaccante ha i mezzi per intercettare i messaggi del sistema 2. L'attaccante ha i mezzi per modificare i messaggi 3. L'attaccante ha i mezzi per spedire il messaggio modificato al destinatario
Postcondizioni	Il sistema rileva il messaggio contraffatto
Scenario principale	<ol style="list-style-type: none"> 1. Il Sistema protegge i messaggi 2. L'attaccante riesce ad intercettare un messaggio e lo modifica 3. Il sistema si accorge del messaggio contraffatto e lo segna nei log
Scenari di un attacco avvenuto con successo	<ol style="list-style-type: none"> 1. Il Sistema protegge i messaggi 2. L'attaccante riesce ad intercettare un messaggio e lo modifica 3. Il sistema accetta il messaggio e agisce di conseguenza, segnando il messaggio nei log

Titolo	ControlloAccessi
Descrizione	L'accesso alle funzionalità del sistema deve essere controllato
Misuse case	FurtoCredenziali, ManInTheMiddle
Relazioni	
Precondizioni	L'attaccante ha i mezzi per carpire in tutto o in parte le credenziali di accesso di un cliente o di un farmacista
Postcondizioni	Il sistema blocca l'accesso non autorizzato e notifica il tentativo di accesso
Scenario principale	<ol style="list-style-type: none"> 1. L'attaccante tenta di accedere al servizio spacciandosi per un utente legittimo, di cui conosce le credenziali solo in parte (ad esempio mediante attacco con dizionario) 2. Il sistema non riconosce le credenziali, restituendo un errore 3. In seguito ad un numero fissato di tentativi falliti, il sistema blocca temporaneamente l'accesso a quell'utente e notifica l'anomalia a chi di dovere
Scenari di un attacco avvenuto con successo	<ol style="list-style-type: none"> 1. L'attaccante riesce a carpire le credenziali di accesso complete di un utente in un qualsiasi modo 2. Il sistema riconosce la correttezza delle credenziali, e fornisce l'accesso al soggetto malevolo 3. L'attaccante ha libero accesso al sistema, con privilegi diversi in base al tipo di utente

2.4.6 Requisiti di Protezione dei Dati

Sussistono inoltre i seguenti requisiti inerenti alla protezione dei dati:

1. I dati salvati devono essere protetti da un attaccante che abbia accesso al sistema, prendendo misure di sicurezza fisica, eventualmente cifrando i dati.
2. I dati inviati tra le parti remote devono essere protetti, utilizzando la cifratura dei dati.
3. Tutte le azioni avvenute sul sistema devono essere tracciate tramite un sistema di log.

La visione e l'analisi dei log verrà gestita con un editor di testo esterno, accessibile solo al personale autorizzato.

ID	Requisiti	Tipo
R16F	Implementazione di un sistema di log per tracciare tutti i messaggi tra i client e i server, inclusi gli accessi, le richieste di prenotazione, di conferma, di sospensione e di invio e ricezione di dati	Funzionale
R9NF	I dati salvati devono essere protetti da un attaccante che abbia accesso al sistema, prendendo misure di sicurezza fisica, eventualmente cifrando i dati	Non Funzionale
R10NF	I dati inviati tra le parti remote devono essere protetti, utilizzando la cifratura dei dati	Non Funzionale

3 Analisi del Problema

3.1 Analisi Documento dei Requisiti: Analisi delle Funzionalità

Tabella delle Funzionalità

Funzionalità	Tipo	Grado di complessità	Requisiti Collegati
Gestione Farmacia	Memorizzazione dati e gestione dati	complessa	R5F, R9F, R10F, R11F, R12F, R13F, R14F, R15F
Registrazione	Interazione esterno e memorizzazione dati	semplice	R4F
RicercaFarmaci	Interazione esterno e lettura dati	semplice	R1F, R2F, R3F
Login	Interazione esterno e lettura dati	semplice	R7F
GestionePrenotazioni	Interazione esterno e memorizzazione dati	comp	R2F, R6F, R8F
ScritturaLog	Memorizzazione dati	semplice	R16F

GestioneFarmacia: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input / Output	Vincoli
Nome Cliente	semplice	Protezione alta	Output	Non più di 40 caratteri
Cognome Cliente	semplice	Protezione alta	Output	Non più di 40 caratteri
Codice Fiscale Cliente	semplice	Protezione media	Output	Deve essere di 16 caratteri
Stato Cliente	semplice	Protezione media	Output	
Lista Farmaci	composto	Protezione alta	Output	
Lista Prenotazioni	composto	Protezione molto alta	Output	

RicercaFarmaci: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input / Output	Vincoli
Nome Farmaco	semplice	Protezione bassa	Input	
Località Utente	composto	Protezione alta	Input	
Farmacia	semplice	Protezione bassa	Input	
Lista Farmacie Pertinenti	composto	Protezione bassa	Output	Non più di 10 farmacie

Registrazione: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Nome Cliente	Semplice	Protezione media	Input	Non più di 40 caratteri
Cognome Cliente	semplice	Protezione media	Input	Non più di 40 caratteri
Data di Nascita	semplice	Protezione media	Input	Deve avere più di 16 anni e data di nascita successiva al 1900
Codice Fiscale	semplice	Protezione media	Input	Deve essere di 16 caratteri
Email	semplice	Protezione alta	Input	Deve essere di 256 caratteri e del formato giusto
Password	semplice	Protezione molto alta	Input	Deve essere almeno di 8 caratteri, di cui uno alfabetico e uno numerico

ScritturaLog: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Data	semplice	Protezione media	Input	Non più di 40 caratteri
Ora	semplice	Protezione media	Input	Non più di 40 caratteri
Attore	semplice	Protezione alta	Input	Non più di 20 caratteri
Identificativo Farmacia	semplice	Protezione alta	Input	Non più di 20 caratteri
Operazione Eseguita	composto	Protezione alta	Input	
Evento	composto	Protezione molto alta	Input	

Login: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Email	semplice	Protezione molto alta	Input	Non più di 256 caratteri
Password	semplice	Protezione molto alta	Input	Non più di 50 caratteri

Gestione Prenotazione: Tabella Informazioni/Flusso

Informazione	Tipo	Livello protezione/privacy	Input/Output	Vincoli
Data invio	semplice	Protezione media	Input	Non più di 40 caratteri
Ora invio	semplice	Protezione media	Input	Non più di 40 caratteri
Data prenotazione	semplice	Protezione media	Input	Solo una data compresa tra il giorno successivo e 14 giorni dopo
Elenco farmaci	composto	Protezione alta	Input	1. Non più di 5 elementi per ogni farmaco 2. Non più di 20 elementi in totale
Identificativo farmacia	semplice	Protezione alta	Input	Non più di 20 caratteri
Identificativo cliente	semplice	Protezione molto alta	Input	Non più di 20 caratteri
Lista prenotazioni	composto	Protezione alta	Output	

3.1.1 Analisi Documento dei Requisiti: Analisi dei Vincoli

Tabella Vincoli

Requisito	Categorie	Impatto	Funzionalità
Semplicità dell'interfaccia	Usabilità	Intuitività di utilizzo	GestioneFarmacia, Registrazione, RicercaFarmaci, Login, NuovaPrenotazione
Velocità della ricerca dei dati	Tempo di Risposta	Maggiore reattività	GestioneFarmacia, Registrazione, RicercaFarmaci, Login, NuovaPrenotazione
Velocità di memorizzazione dei dati	Tempo di Risposta	Maggiore reattività	GestioneFarmacia, Registrazione, Login, NuovaPrenotazione
Controllo Accessi	Sicurezza	Peggiorano tempo di risposta e usabilità, migliorano la privacy dei dati	GestioneFarmacia, NuovaPrenotazione
Protezione dei Dati	Sicurezza	Peggiorano tempo di risposta, migliorano la privacy dei dati	GestioneFarmacia, Registrazione, RicercaFarmaci, Login, NuovaPrenotazione

3.1.2 Analisi Documento dei Requisiti: Analisi delle Interazioni

Tabella Maschere

Maschera	Informazioni	Funzionalità
Home Gestione	messaggio di benvenuto e scelta della funzionalità	GestioneFarmacia
View Login	email, password	Login
View Prenotazioni	lista prenotazioni	GestioneFarmacia
View ResocontoUtenti	nome cliente, cognome cliente, codice fiscale cliente, stato cliente	GestioneFarmacia
View VerificaIdentità	nome cliente, cognome cliente, codice fiscale cliente	VerificaIdentità
View Farmaci	lista farmaci	gestioneFarmacia
Home Servizio	messaggio di benvenuto, nome farmaco, località utente, lista farmacie pertinenti	RicercaFarmaci
View Registrazione	nome cliente, cognome cliente, data di nascita, codice fiscale, email, password	Registrazione
View NuovaPrenotazione	data invio, ora invio, data prenotazione, elenco farmaci, identificativo farmacia, identificativo cliente	NuovaPrenotazione
View Prenotazioni-Personali	lista prenotazioni	ListaPrenotazioni

Tabella Sistemi Esterni

Sistema	Descrizione	Protocollo di Interazione	Livello di Sicurezza
Gestione Magazzino	Sistema che si occupa della gestione dei farmaci in magazzino	GestioneMagazzino mette a disposizione delle funzionalità di elencazione dei farmaci	Medio livello di sicurezza perchè protegge i dati della farmacia

3.1.3 Analisi Ruoli e Responsabilità

Tabella Ruoli

Ruolo	Responsabilità	Maschere	Riservatezza	Numerosità
Farmacista	Gestione di tutte le informazioni relative agli utenti e alle prenotazioni di una farmacia	Home Gestione, View Login, View Prenotazioni, View ResocontoUtenti, View VerificaIdentità, View Farmaci,	È richiesto un alto grado di riservatezza	Massimo 10 farmacisti per ogni farmacia
Cliente	Ricerca di un farmaco senza necessità di login	Home Servizio, View Login, View Registrazione	È richiesto un medio grado di riservatezza	Illimitati
ClienteRegistrato	Ricerca e prenotazione di farmaci presso una farmacia	Home Servizio, View NuovaPrenotazione, View PrenotazioniPersonal	È richiesto un alto grado di riservatezza	Illimitati

Farmacista: Tabella Ruolo-Informazioni

Informazione	Tipo di Accesso
Nome Cliente	Lettura
Cognome Cliente	Lettura
Codice Fiscale	Lettura
Stato Cliente	Lettura/Scrittura
Lista Farmaci	Lettura/Scrittura
Lista Prenotazioni	Lettura/Scrittura

ClienteRegistrato: Tabella Ruolo-Informazioni

Informazione	Tipo di Accesso
Nome Cliente	Lettura/Scrittura
Cognome Cliente	Lettura/Scrittura
Data di Nascita	Lettura
Codice Fiscale	Lettura
Email	Lettura/Scrittura
Password	Lettura/Scrittura
Nome Farmaco	Scrittura
Località Utente	Lettura
Lista Farmacie Pertinenti	Lettura
Data prenotazione	Scrittura
Elenco farmaci	Scrittura

Cliente: Tabella Ruolo-Informazioni

Informazione	Tipo di Accesso
Nome Farmaco	Scrittura
Località Utente	Scrittura
Lista Farmacie Pertinenti	Lettura

3.1.4 Scomposizione del Problema

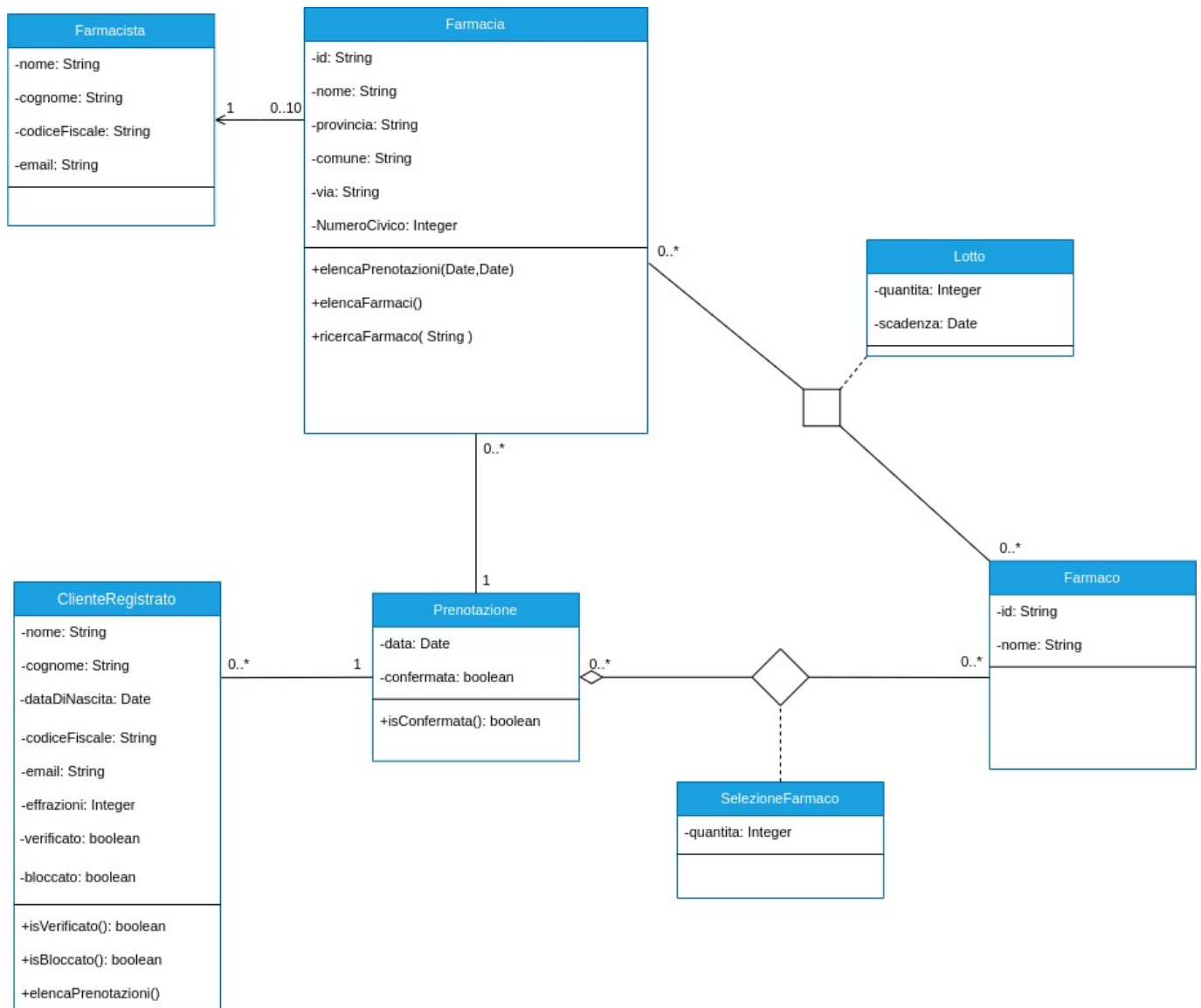
Tabella Scomposizione Funzionalità

Funzionalità	Scomposizione
GestioneFarmacia	ResocontoFarmaci, ResocontoUtenti, ControlloPrenotazioni, VerificaIdentità
GestionePrenotazioni	NuovaPrenotazione, ListaPrenotazioni
ControlloPrenotazioni	ConfermaPrenotazione
ResocontoUtenti	SospensioneUtenza

Non sono presenti legami di esclusione o di necessità tra le sotto-funzionalità del sistema.

3.1.5 Creazione Modello del Dominio

Il seguente diagramma delle classi rappresenta la parte di modello del dominio relativa al sistema.



3.1.6 Architettura Logica: Struttura

Diagramma dei package

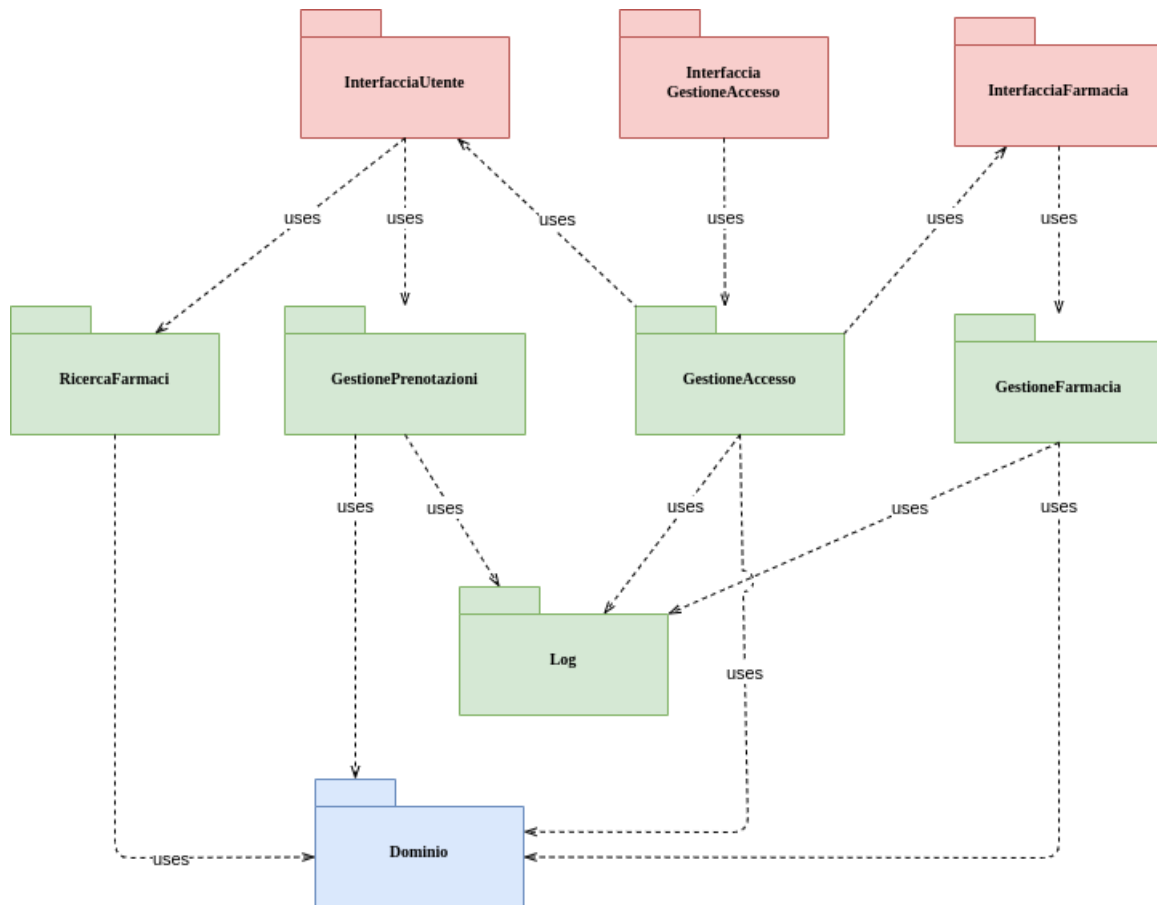


Diagramma delle classi: InterfacciaGestioneAccesso & GestioneAccesso

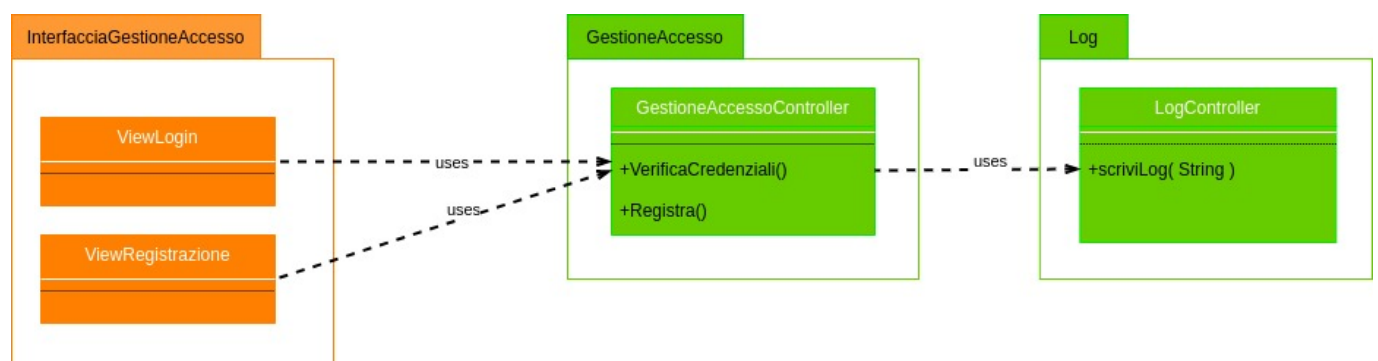


Diagramma delle classi: InterfacciaGestioneFarmacia & GestioneFarmacia

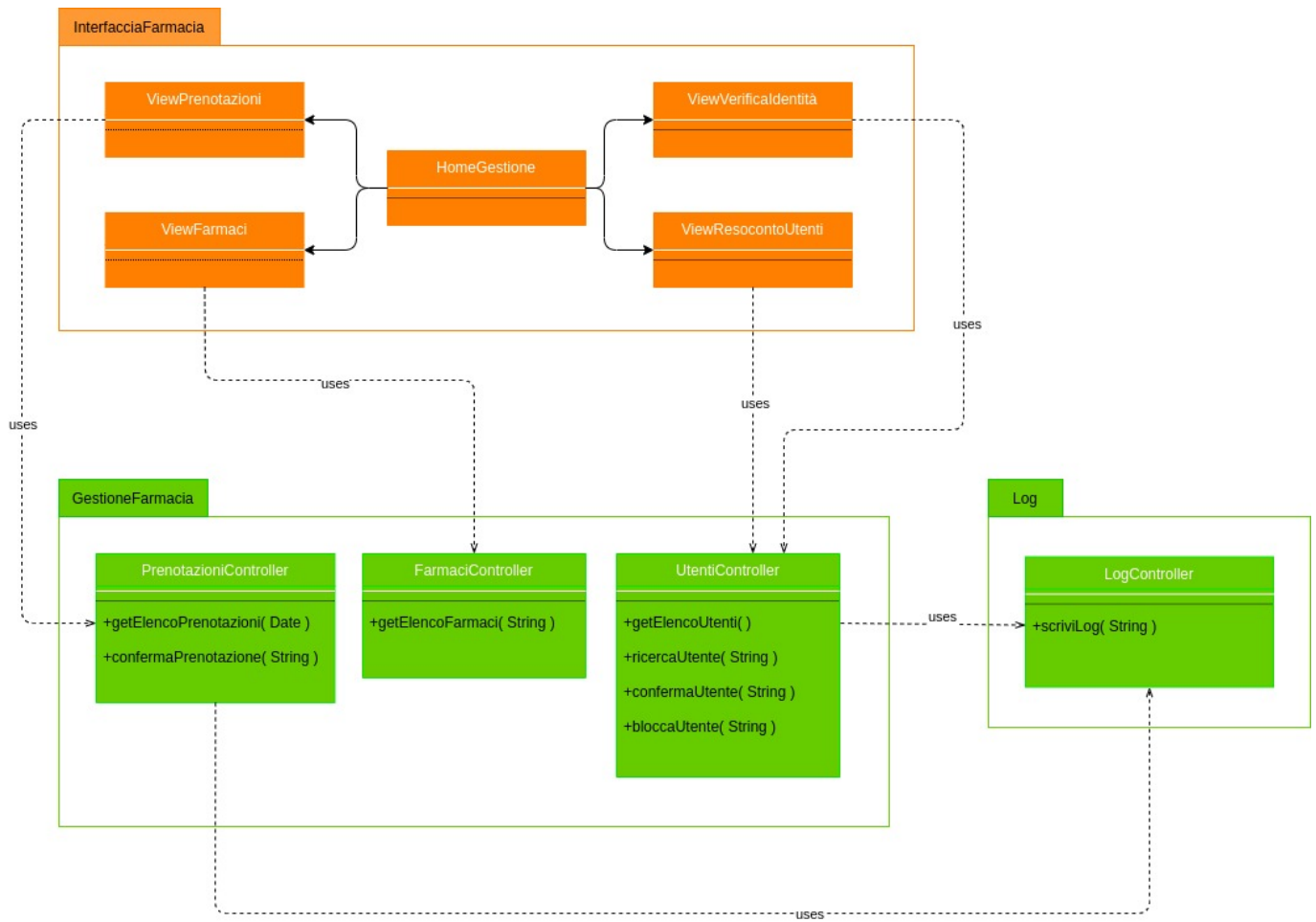
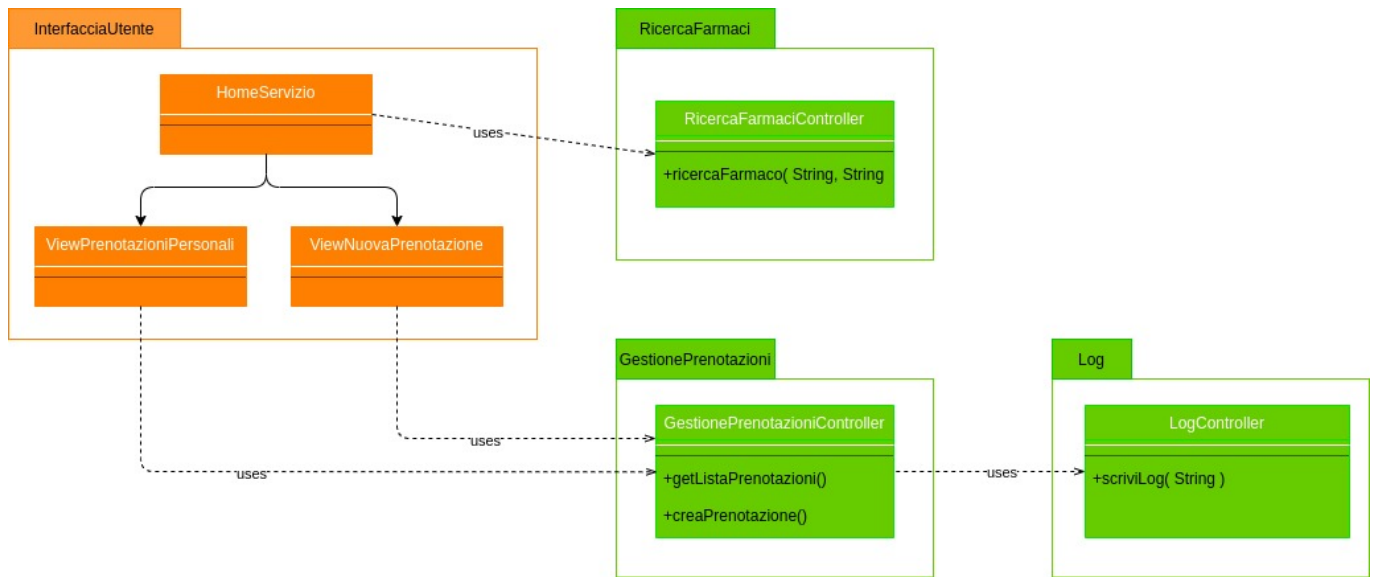


Diagramma delle classi: InterfacciaUtente & RicercaFarmaci & GestionePrenotazioni



3.1.7 Architettura Logica: Interazione

Diagramma di Sequenza: Login Utente

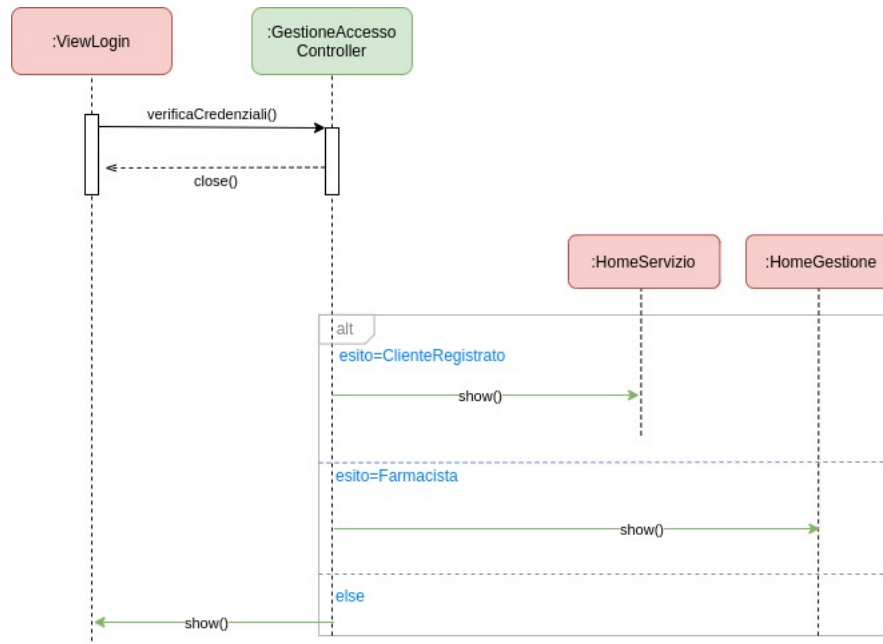


Diagramma di Sequenza: Registrazione Utente

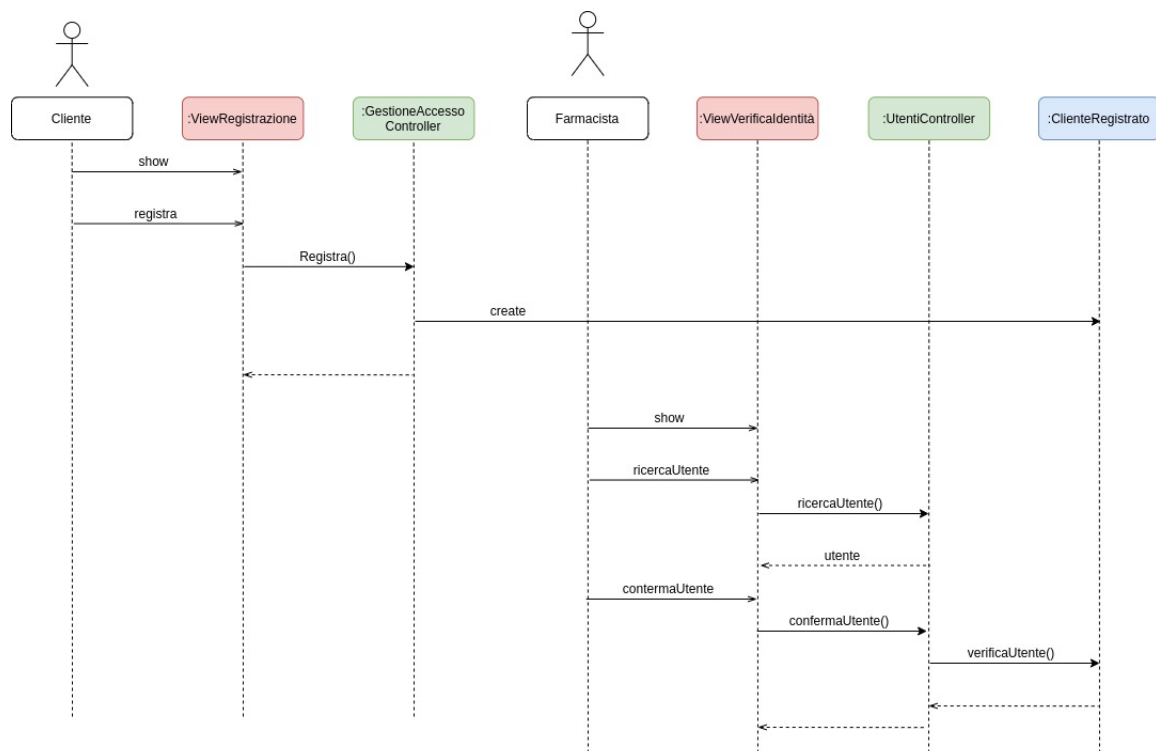


Diagramma di Sequenza: Nuova Prenotazione

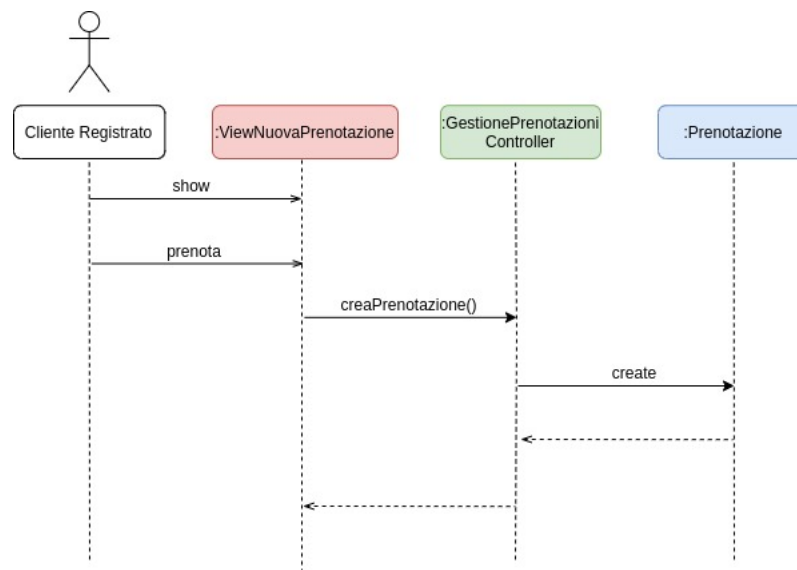


Diagramma di Sequenza: Conferma Prenotazione

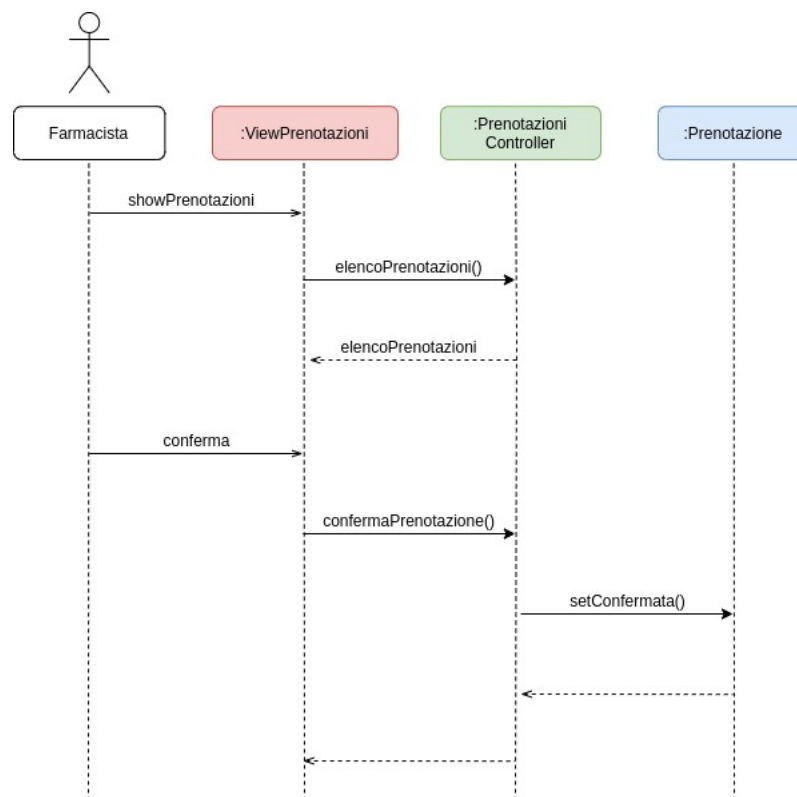


Diagramma di Sequenza: Ricerca Farmaco

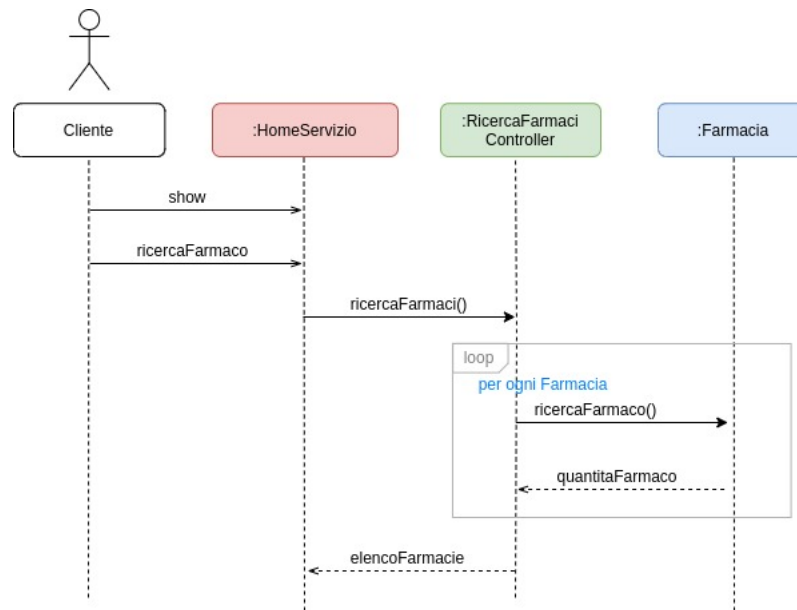
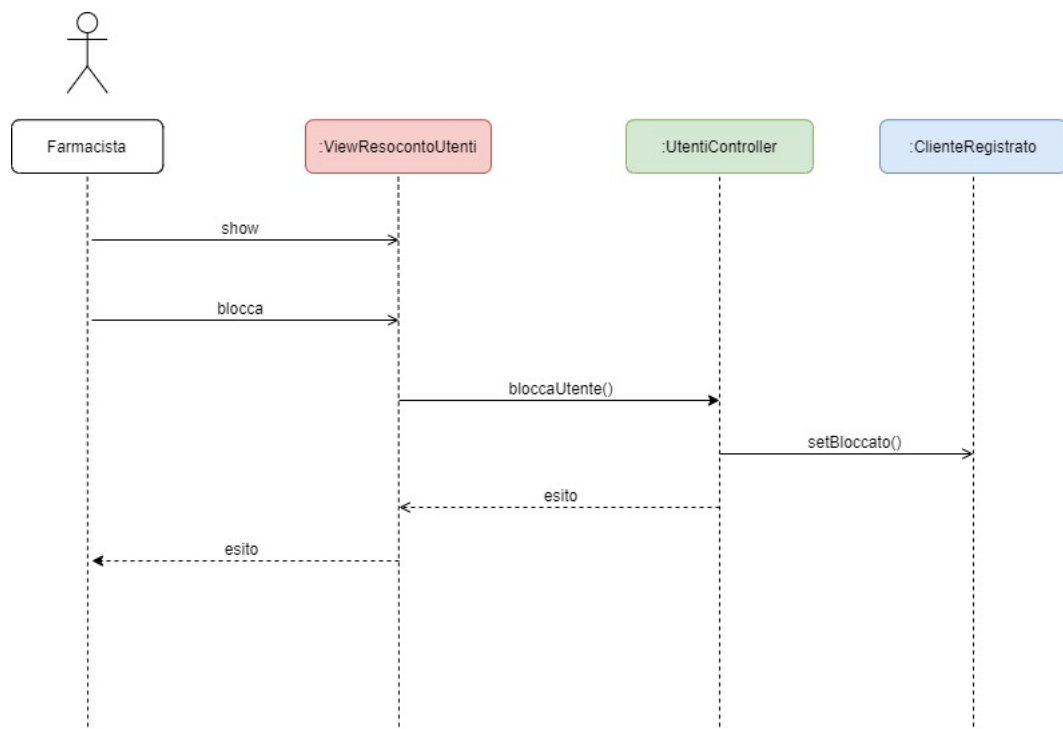
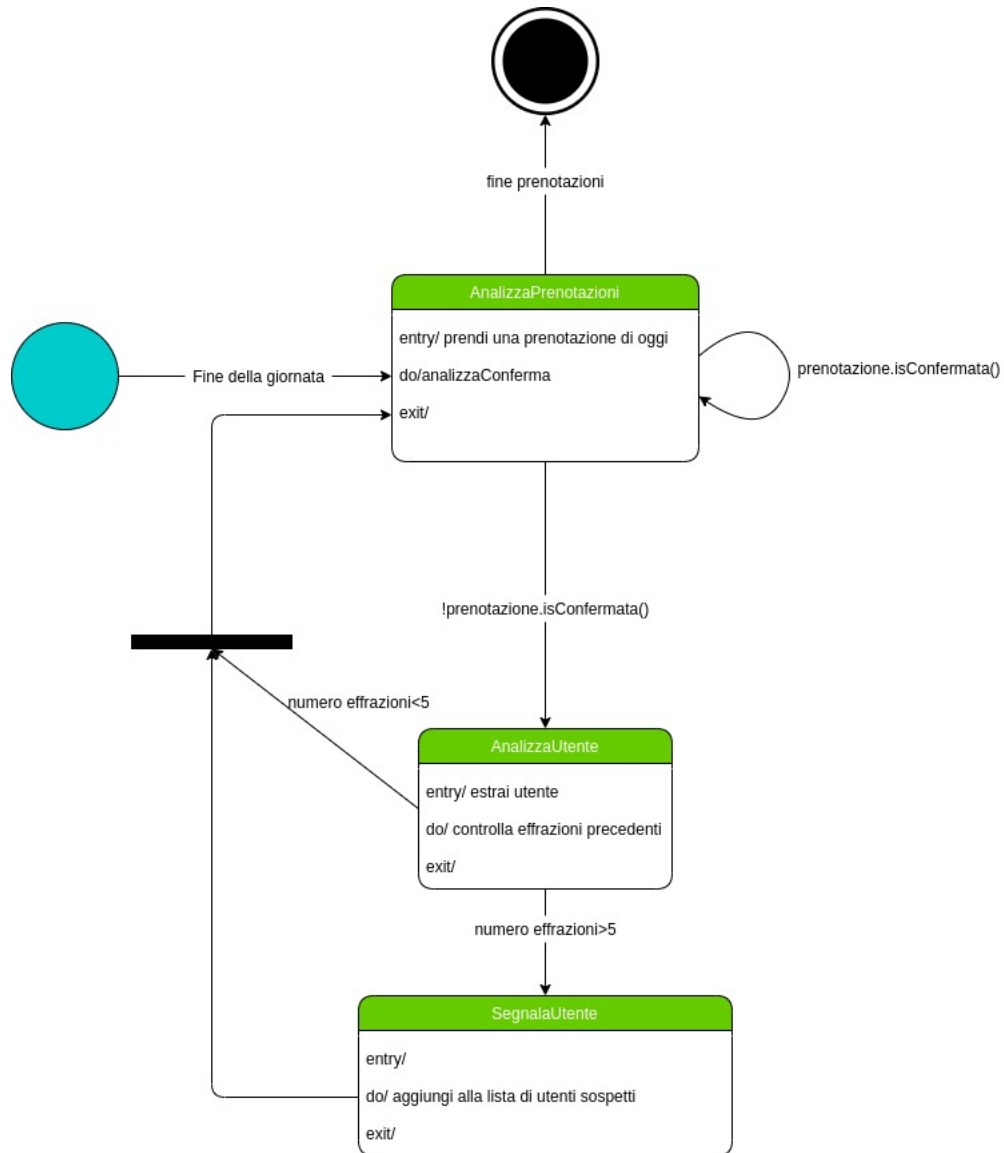


Diagramma di Sequenza: Sospensione Utente



3.1.8 Architettura Logica: Comportamento

Diagramma di Stato: Analizza Utente



3.1.9 Piano di Lavoro

I compiti sono stati divisi in base alle competenze di ogni membro del gruppo come indicato nella tabella sottostante:

Package	Progetto	Sviluppo
Dominio	Guerra, Palaferri, Romanini	Guerra
Log	Guerra, Palaferri, Romanini	Guerra
RicercaFarmaci	Guerra, Palaferri, Romanini	Palaferri
GestionePrenotazioni	Guerra, Palaferri, Romanini	Palaferri, Romanini
GestioneAccesso	Guerra, Palaferri, Romanini	Guerra, Romanini
GestioneFarmacia	Guerra, Palaferri, Romanini	Palaferri, Romanini
InterfacciaUtente	Guerra, Palaferri, Romanini	Romanini
InterfacciaGestioneAccsso	Guerra, Palaferri, Romanini	Guerra
InterfacciaFarmacia	Guerra, Palaferri, Romanini	Palaferri, Romanini

I tempi di rilascio sono i seguenti:

- Progettazione entro due settimane dalla data odierna
- Sviluppo dei vari moduli con annessi test unitari entro una settimana dalla fine della fase di progettazione
- Integrazione e testing del sistema entro una settimana dalla fine dello sviluppo

Sviluppi Futuri

Il cliente ha richiesto la creazione di un applicativo mobile per sistemi android e iOS, con l'obiettivo di rendere il più pratico possibile l'utilizzo del programma.

3.1.10 Piano del Collaudo

```
1 public class testPrenotazione{
2     private Prenotazione prenotazione;
3
4     @Before
5     public void setUp(){
6         prenotazione = new Prenotazione();
7     }
8
9     @Test
10    public void testCostruttore(){
11        prenotazione = new Prenotazione(new SimpleDateFormat("2021-06-01"), true);
12        Assert.assertNull(prenotazione.isConfermata());
13    }
14
15    @Test
16    public void testGetter(){
17        prenotazione = new Prenotazione(new SimpleDateFormat("2021-06-01"), true);
18        Assert.assertEquals(prenotazione.getData(), new SimpleDateFormat("2021-06-01"));
19        Assert.assertEquals(prenotazione.getConfermata(), true);
20    }
21
22    @Test
23    public void testSetter(){
24        prenotazione.setData(new SimpleDateFormat("2021-07-02"));
25        Assert.assertEquals(prenotazione.getData(), new SimpleDateFormat("2021-07-02"));
26        prenotazione.setConfermata(true);
27        Assert.assertEquals(prenotazione.getConfermata(), true);
28    }
29 }
30
31 public class testFarmacia{
32     private Farmacia farmacia;
33
34     @Before
35     public void setUp(){
36         farmacia = new Farmacia();
37     }
38
39     @Test
40     public void testCostruttore(){
41         Assert.assertNull(farmacia.getNome());
42         Assert.assertNull(farmacia.getId());
43     }
44
45     @Test
46     public void testGetter(){
```

```

47     farmacia = new Farmacia("N23N230SD", "Ubertini", "BO", "Bologna", "via Libia", "10");
48     Assert.assertEquals(farmacia.getId(), "N23N230SD");
49     Assert.assertEquals(farmacia.getNome(), "Ubertini");
50     Assert.assertEquals(farmacia.getProvincia(), "BO");
51     Assert.assertEquals(farmacia.getComune(), "Bologna");
52     Assert.assertEquals(farmacia.getVia(), "via Libia");
53     Assert.assertEquals(farmacia.getNumeroCivico(), "10");
54 }
55
56 @Test
57 public void testSetter(){
58     farmacia.setId("N23N230SD");
59     Assert.assertEquals(farmacia.getId(), "N23N230SD");
60     farmacia.setNome("Ubertini");
61     Assert.assertEquals(farmacia.getNome(), "Ubertini");
62     farmacia.setProvincia("BO");
63     Assert.assertEquals(farmacia.getProvincia(), "BO");
64     farmacia.setComune("Bologna");
65     Assert.assertEquals(farmacia.getComune(), "Bologna");
66     farmacia.setVia("via Libia");
67     Assert.assertEquals(farmacia.getVia(), "via Libia");
68     farmacia.setNumeroCivico("10");
69     Assert.assertEquals(farmacia.getNumeroCivico(), "10");
70 }
71 }
72
73 public class testCliente{
74     private ClienteRegistrato cliente;
75
76     @Before
77     public void setUp(){
78         cliente = new ClienteRegistrato();
79     }
80
81     @Test
82     public void testGetter(){
83         cliente = new ClienteRegistrato("Federico", "Chesani", new
84             SimpleDateFormat("1920-07-10"), "CHSFRC20L10A944G",
85             "federico.chesani@unibo.it", 0, null, true, false);
86         Assert.assertEquals(cliente.getNome(), "Federico");
87         Assert.assertEquals(cliente.getCognome(), "Chesani");
88         Assert.assertEquals(cliente.getNascita(), new SimpleDateFormat("1920-07-10"));
89         Assert.assertEquals(cliente.getCodiceFiscale(), "CHSFRC20L10A944G");
90         Assert.assertEquals(cliente.getEmail(), "federico.chesani@unibo.it");
91         Assert.assertEquals(cliente.getEffrazioni(), 0);
92         Assert.assertEquals(cliente.isVerificato(), true);
93         Assert.assertEquals(cliente.isBloccato(), false);
94     }

```

```
95
96  @Test
97  public void testSetter(){
98      cliente.setNome("Federico");
99      Assert.assertEquals(cliente.getNome(), "Federico");
100     cliente.setCognome("Chesani");
101     Assert.assertEquals(cliente.getCognome(), "Chesani");
102     cliente.setNascita(new SimpleDateFormat("1920-07-10"));
103     Assert.assertEquals(cliente.getNascita(), new SimpleDateFormat("1920-07-10"));
104     cliente.setCodiceFiscale("CHSFRC20L10A944G");
105     Assert.assertEquals(cliente.getCodiceFiscale(), "CHSFRC20L10A944G");
106     cliente.setEmail("federico.chesani@unibo.it");
107     Assert.assertEquals(cliente.getEmail(), "federico.chesani@unibo.it");
108     cliente.setEffrazioni(0);
109     Assert.assertEquals(cliente.getEffrazioni(), 0);
110     cliente.setVerificato(true);
111     Assert.assertEquals(cliente.isVerificato(), true);
112     cliente.setBloccato(false);
113     Assert.assertEquals(cliente.isBloccato(), false);
114 }
115 }
```


4 Progettazione

4.1 Progettazione Architettuale

4.1.1 Requisiti non funzionali

Dall'analisi dei requisiti sono emersi i seguenti requisiti non funzionali:

- Tempo di risposta
- Usabilità
- Integrità dei dati
- Protezione dei dati
- Sicurezza delle comunicazioni

La protezione dei dati e delle comunicazioni assume fondamentale importanza vista la natura del software, che deve trattare dati personali e sanitari dei clienti. La compromissione di questi risulterebbe in una grave perdita finanziaria e di immagine, senza considerare i danni apportati alla privacy degli utenti. Inoltre, sarà necessario assicurare la sicurezza fisica dei dati immagazzinati nel sistema. L'introduzione di misure di sicurezza delle comunicazioni e protezione dei dati non compromette l'usabilità del sistema, ma potrebbe peggiorarne leggermente le prestazioni: è possibile comunque bilanciare le due esigenze senza eccessive complicazioni mediante le tecnologie esposte in seguito. Va notato inoltre che il sistema non presenta vincoli di tempo particolarmente stringenti (nessun vincolo real-time).

4.1.2 Scelte tecnologiche

La scelta tecnologica principale ricade sul tipo di applicazione che si andrà a sviluppare. In questo caso la scelta è stata quella di sviluppare un'applicazione web, per vari motivi: prima di tutto, consente di avere una piattaforma standard accessibile da quasi tutti i dispositivi, con il solo requisito di un browser web. In questo modo si evita di restringere le possibilità di accesso al servizio. Inoltre, un'applicazione web consente di avere una gestione maggiormente centralizzata ed un deployment più agevole (a questo proposito si veda la sezione apposita del deployment).

4.1.3 Scelta dell'architettura

Dopo una rapida analisi, si è constatato che l'architettura più adeguata per il sistema è l'**architettura client-server a 3 livelli**.

L1 – Client

La componente lato Client sarà implementata da due interfacce differenti:

- Un'interfaccia per le funzionalità relative ai clienti (registrati e non)
- Un'interfaccia per la gestione della farmacia da parte di un operatore (farmacista)

L2 – Server

Rispettando il principio del "minimo privilegio" per limitare i danni in caso di attacco e per distribuire meglio il carico, si è deciso di scomporre i server in base alle funzionalità offerte. Si hanno quindi tre server:

- Un server che fornisce i servizi ai clienti registrati e non
- Un server che funge da pannello di controllo per i farmacisti
- Un server per le funzionalità di autenticazione

L3 – Persistenza

La gestione della persistenza verrà implementata in un server dedicato sul quale sarà installato un DBMS che gestisca i dati di tutte le farmacie aderenti al servizio. Su tale server sarà installato il DBMS IBM DB2. L'interfacciamento con il DBMS avverrà mediante la metodologia "forza bruta" utilizzando i metodi CRUD. Per quanto riguarda il log delle operazioni, invece, questo verrà salvato su file system (un semplice file sul server adibito)

4.1.4 Pattern architetturali e di design

Infine, dopo un'attenta analisi, si è optato per l'adozione del pattern **Broker**: un componente verrà interposto alla comunicazione Client–Server e avrà il compito di indirizzare le richieste dei client al relativo server, effettuando un controllo sulle sessioni attive per determinare lo stato del client. La scomposizione in diversi client e server consente di avere una separazione netta tra gli applicativi del cliente e del farmacista, in modo da localizzare le operazioni critiche e ottenere maggiore protezione dei dati. Il pattern Model View Controller (**MVC**) è stato invece scelto come pattern architetturale. Chiaramente l'affidabilità del sistema dipende dalla robustezza del broker e soprattutto del sistema di autenticazione.

Si riportano di seguito i diagrammi di package e componenti che descrivono l'architettura del sistema.

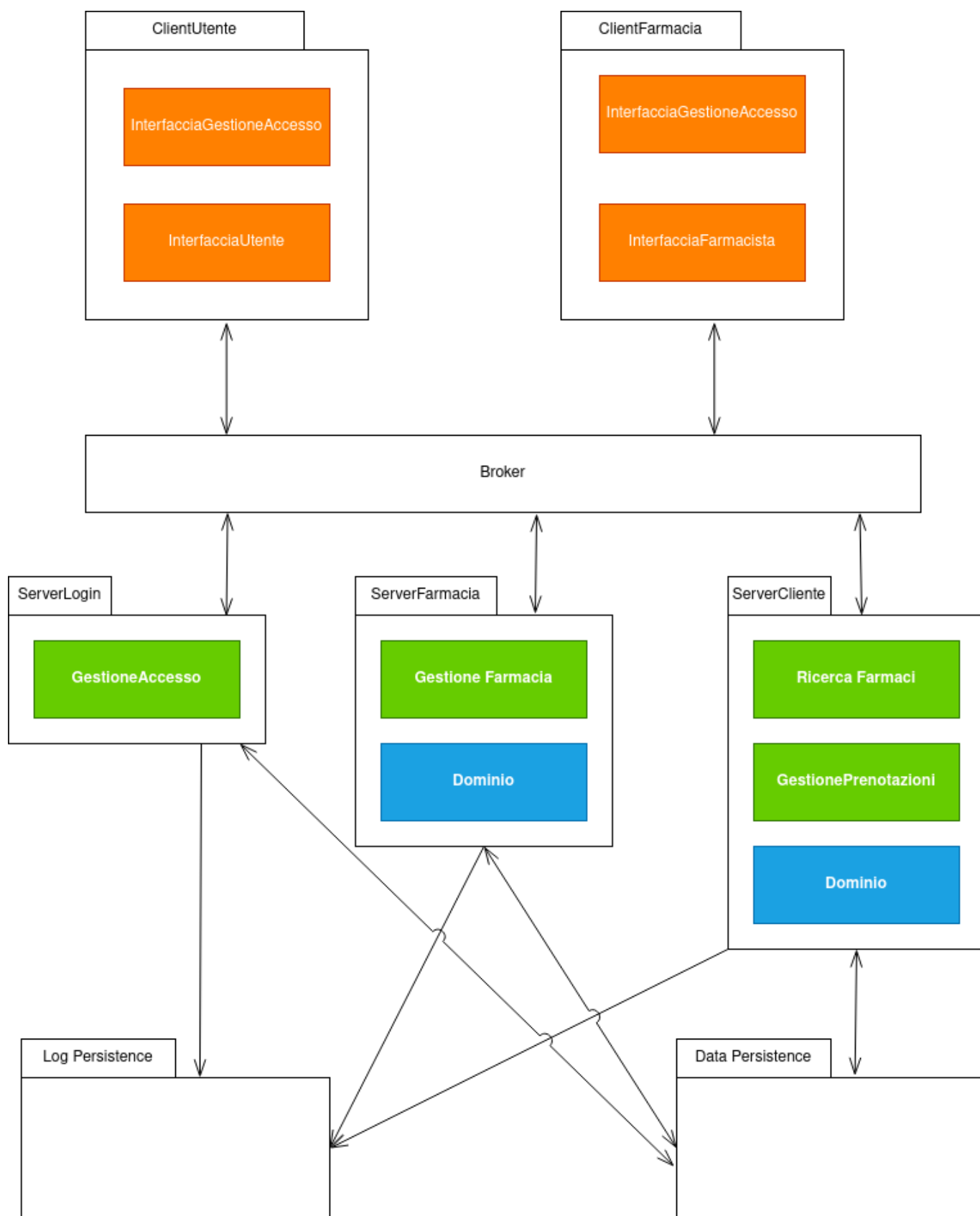


Figura 1: Diagramma dei package

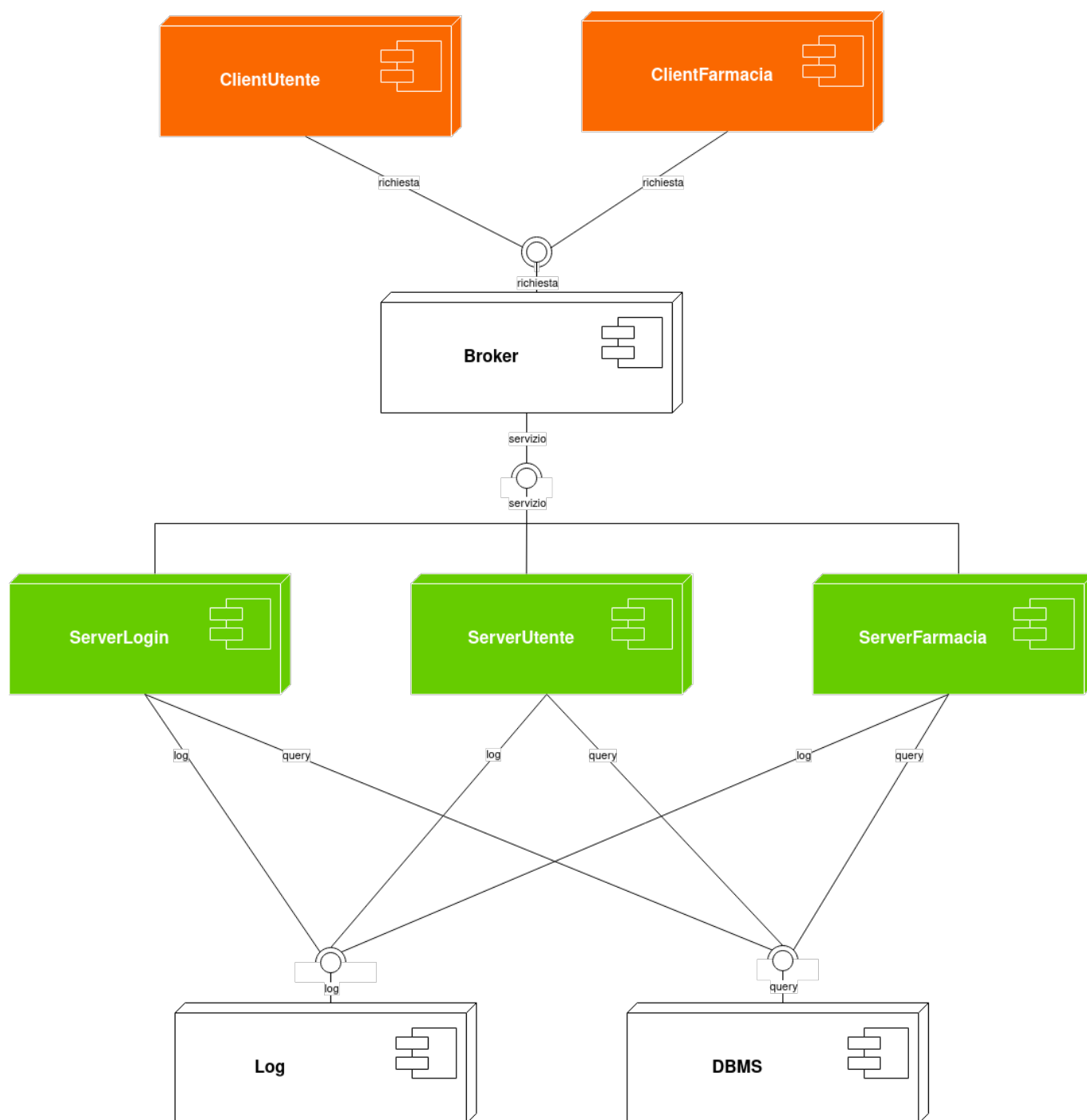


Figura 2: Diagramma dei componenti

4.2 Progettazione di dettaglio

4.2.1 Struttura

Struttura: Dominio

Per quanto riguarda il dominio, i diagrammi rimangono sostanzialmente uguali a quelli visti in analisi. Nonostante il dominio del cliente sia pressoché identico a quello del farmacista, si è comunque deciso di distinguere i due domini al fine di evitare l'introduzione di classi non necessarie. In particolare, la parte di applicativo relativa al cliente non dovrà gestire né conoscere i farmacisti legati ad ogni farmacia (informazione nota solo al server delle farmacie).

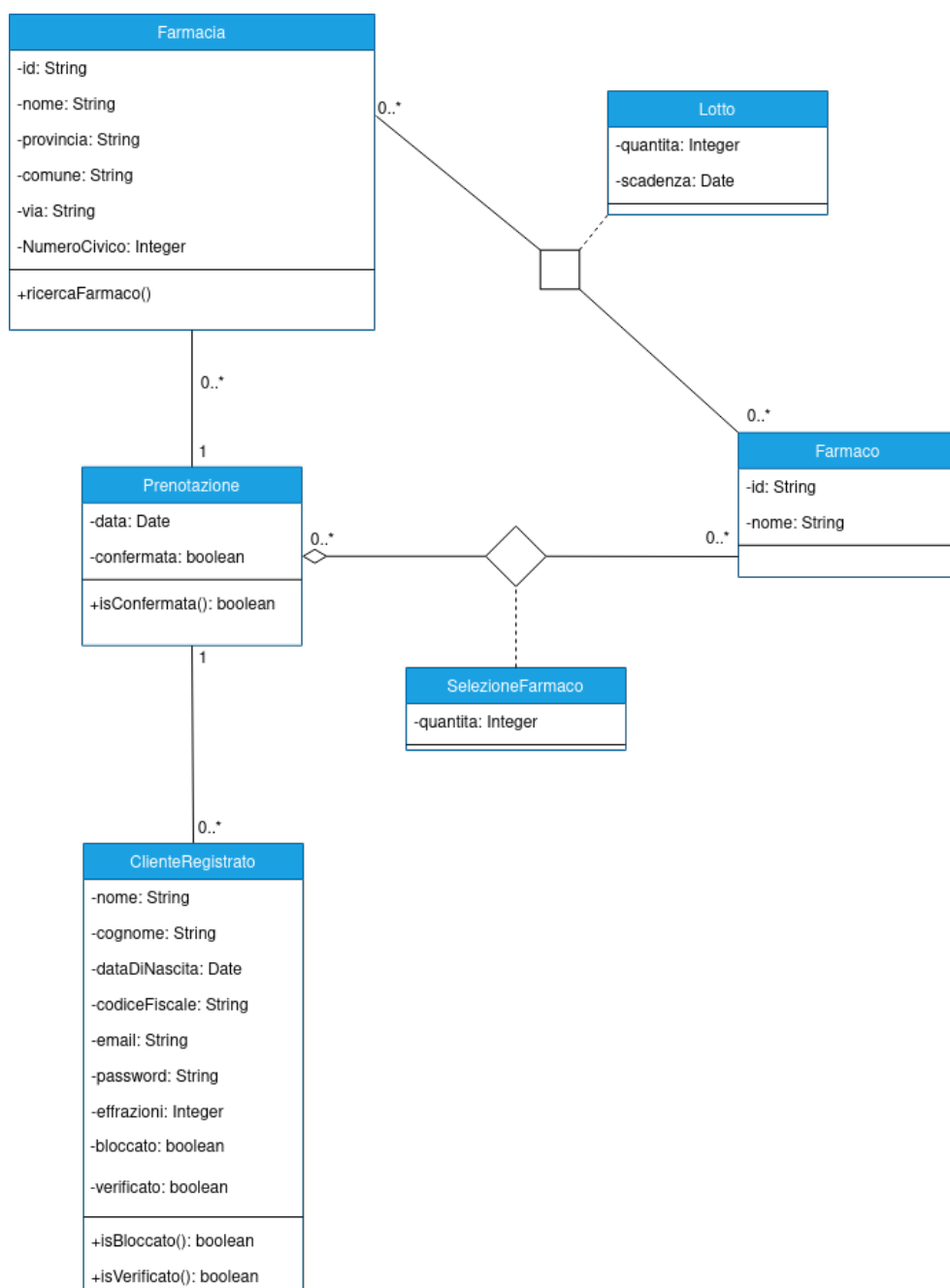


Figura 3: Diagramma di dettaglio: Dominio Clienti

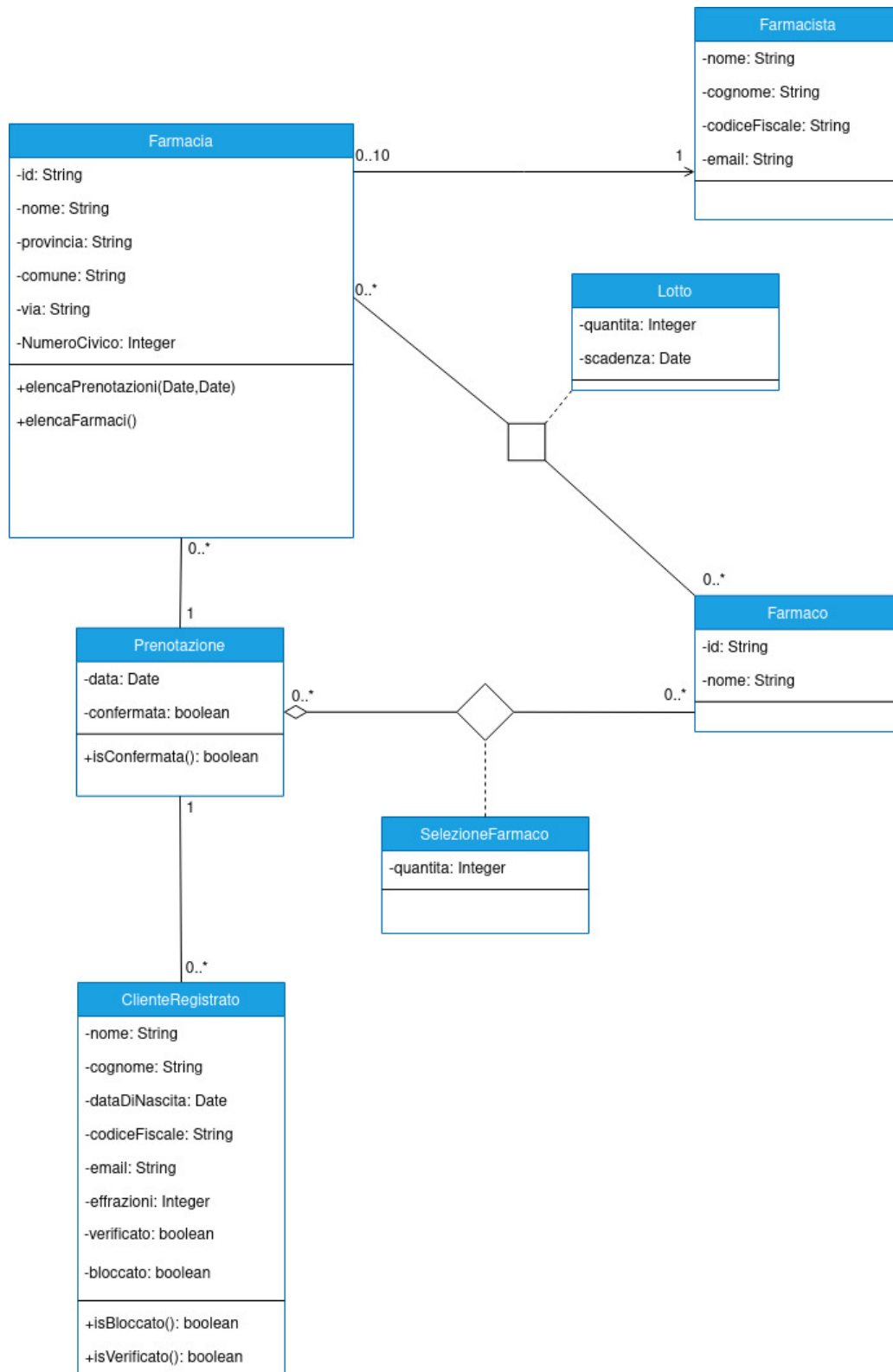


Figura 4: Diagramma di dettaglio: Dominio Farmacia

Come si può notare, il dominio della farmacia presenta diversi metodi aggiuntivi (in particolare per effettuare operazioni sul cliente e per elencare farmaci) non necessari al cliente e anzi da nascondere ad esso per evitare che i permessi o privilegi vengano aggirati. Inoltre, le associazioni **Lotto** e **SelezioneFarmaco** sono state mantenute nei diagrammi per chiarezza. L'associazione **Lotto** dovrà necessariamente essere una classe a sé stante, mentre l'associazione **SelezioneFarmaco** potrà essere concretizzata in classe o sostituita da un semplice Integer. Quest'ultima scelta è lasciata agli implementatori. Le associazioni vere e proprie dovranno poi essere implementate mediante una **mappa**, ad esempio con un oggetto del tipo `HashMap<Farmaco, Lotto>`

Struttura: Interfacce

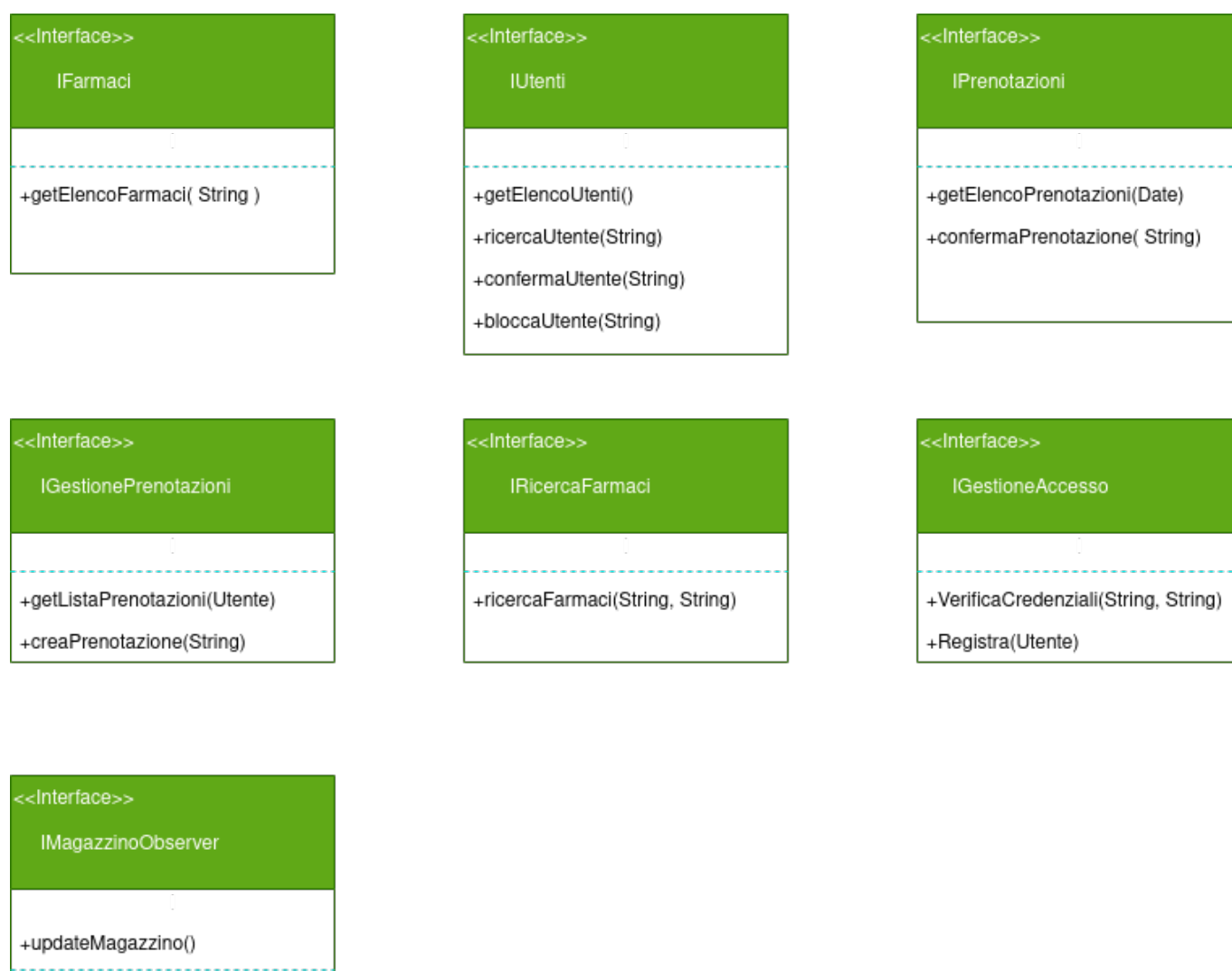


Figura 5: Diagramma di dettaglio: Interfacce

L'aggiunta di tali interfacce consente di applicare il *Dependency Inversion Principle* in modo da disaccoppiare gli utilizzatori dalle implementazioni, che potrebbero cambiare. L'interfaccia **IMagazzinoObserver** è stata introdotta per l'utilizzo di un pattern Observer, i cui dettagli vengono esposti nel prossimo paragrafo.

Struttura: Controller

Si è deciso di utilizzare una classe **Controller** in cui inserire le funzionalità relative alla persistenza (Database e Log). Si è pensato di posizionare questa classe in cima alla gerarchia dei controller: in questo modo, le funzionalità comuni di lettura/scrittura su database e log sono riutilizzabili dai controller figli senza bisogno di reimplementarle. Nonostante il controller "monolitico" non rispetti il *Single Responsibility Principle*, abbiamo comunque optato per questa soluzione, in quanto le funzionalità relative al database e al logging risultano facilmente accoppiabili essendo entrambe legate a un qualche tipo di persistenza. Inoltre, non si prevede alcun tipo di estensione/modifica per quanto riguarda la persistenza.

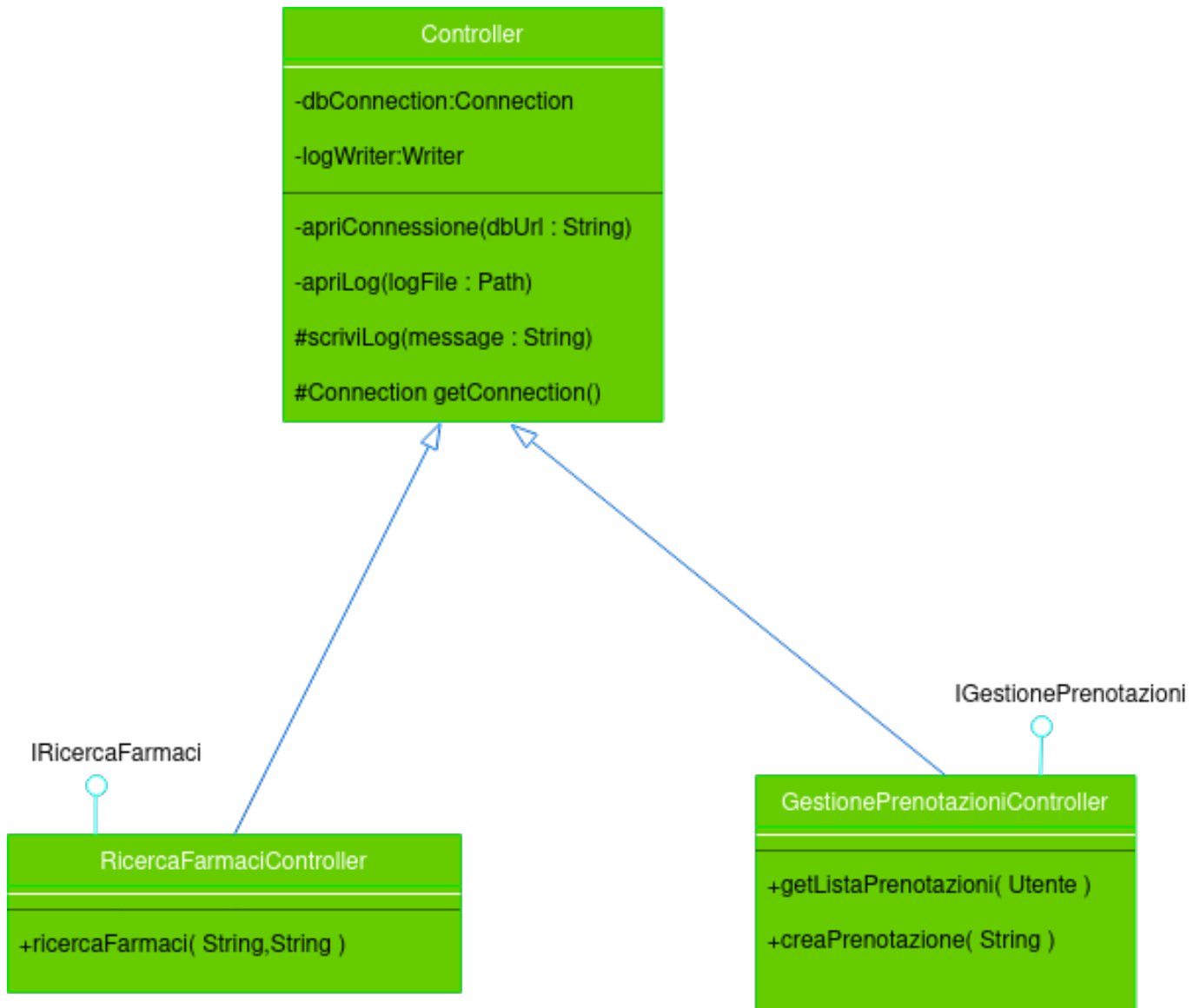


Figura 6: Diagramma di dettaglio: RicercaFarmaci, GestionePrenotazioni

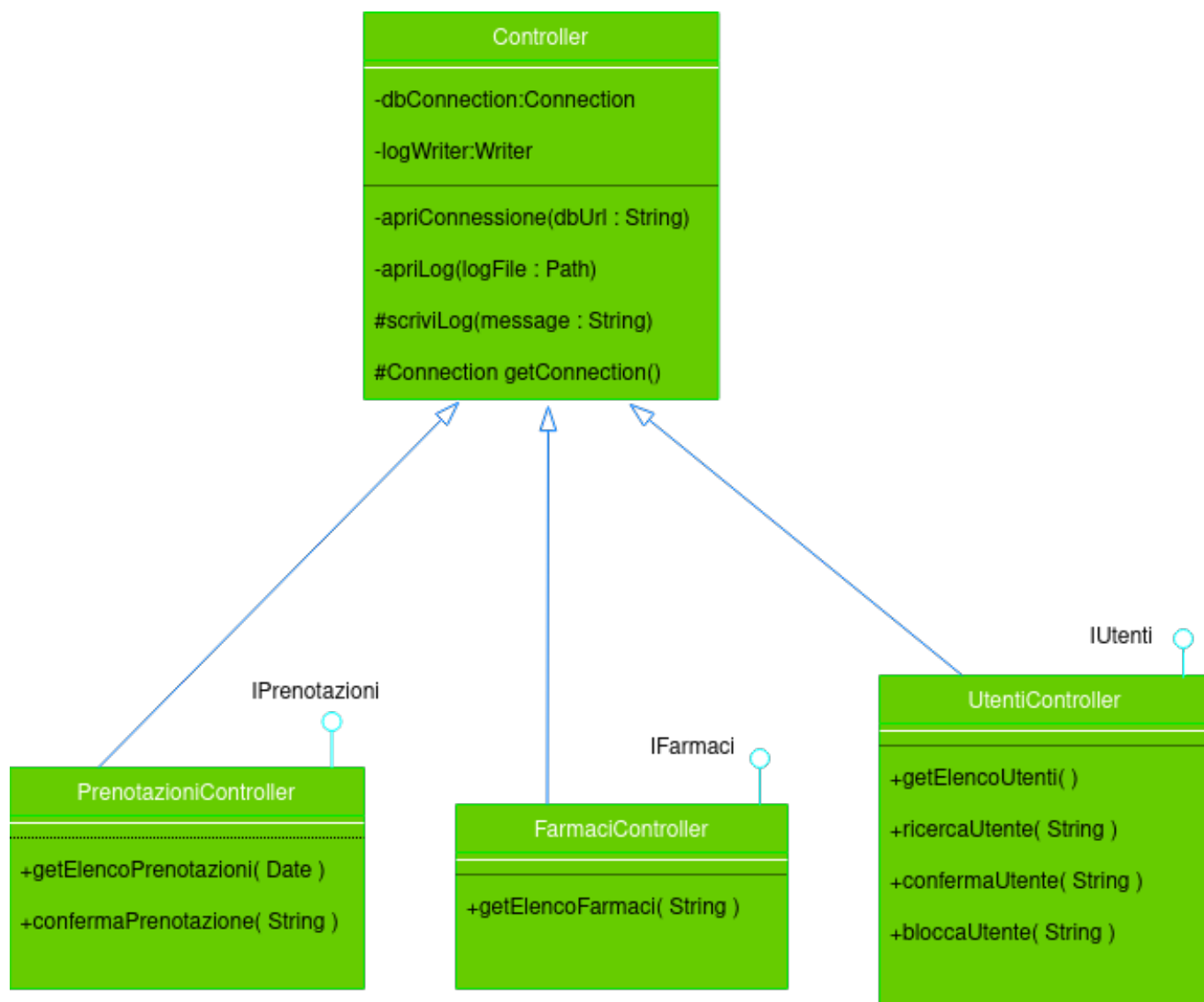


Figura 7: Diagramma di dettaglio: GestioneFarmacia

Anche i controller presenti sul server relativo alle farmacie seguono lo stesso principio esposto sopra.

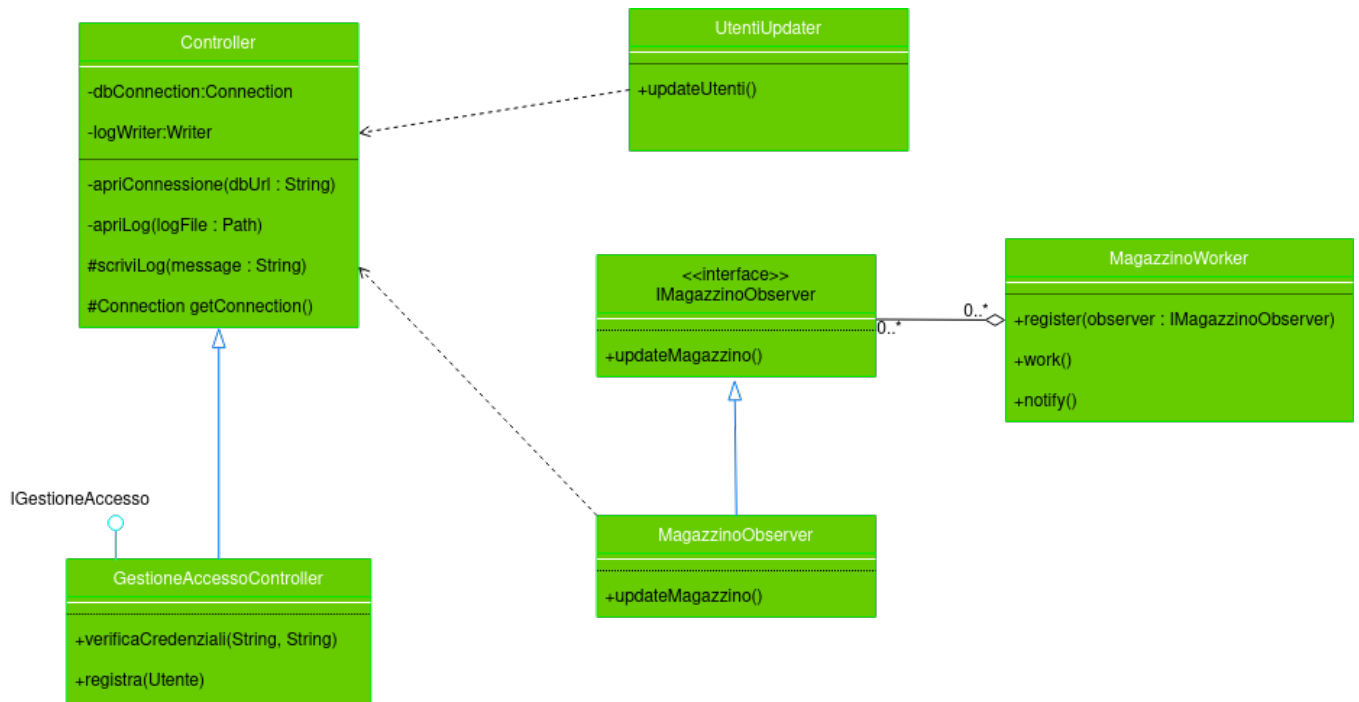


Figura 8: Diagramma di dettaglio: GestioneAccesso

Il diagramma di dettaglio del server login risulta più complesso degli altri due poiché contiene le ulteriori classi necessarie per implementare l'aggiornamento dello stato dei clienti e caching del database in seguito al ricevimento di eventi. In particolare, per implementare il caching locale del database remoto contenente i dati dei magazzini, si è deciso di utilizzare un pattern *Observer*: Un'istanza della classe **MagazzinoObserver** viene registrata nel **MagazzinoWorker**. Quest'ultimo poi comunicherà con il server remoto mediante un protocollo prestabilito e, alla ricezione di un aggiornamento da parte del server, notificherà l'Observer. Lo scambio dei dati riguardanti l'aggiornamento può avvenire tra Worker e Observer in diversi modi, pertanto la scelta viene lasciata agli implementatori.

Si noti che la scelta del pattern Observer è dettata dal fatto che l'evento di aggiornamento del database remoto può risultare importante anche per future estensioni del software: per questo motivo l'Observer si basa sull'interfaccia **IMagazzinoObserver**.

Per quanto riguarda invece l'aggiornamento dello stato dei clienti a fine giornata, il metodo `updateUtenti()` della classe **UtentiUpdater** verrà invocato automaticamente allo scattare di un nuovo giorno, in base all'orario del server.

Struttura: Broker

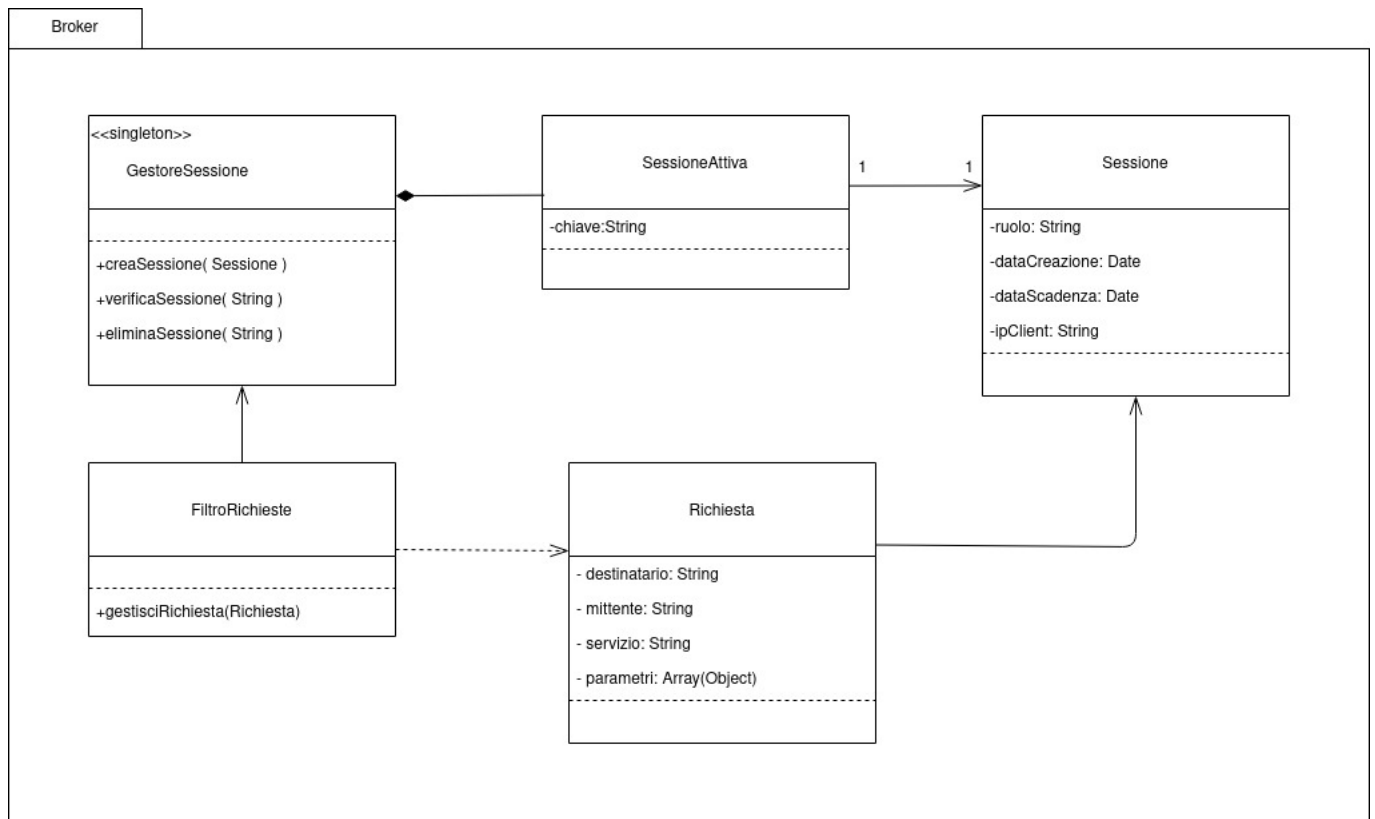


Figura 9: Diagramma di dettaglio: Broker

Struttura: View

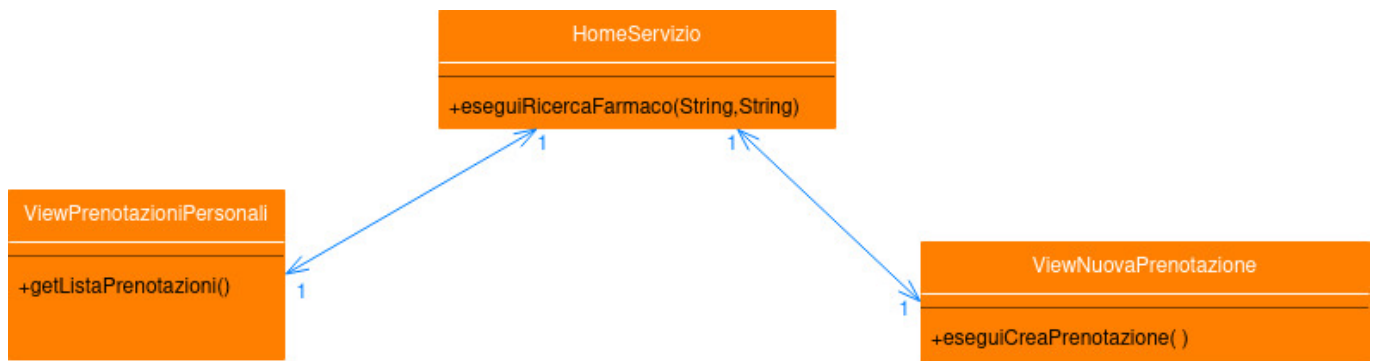


Figura 10: Diagramma di dettaglio: InterfacciaUtente

Inserire disegni interfacce qui

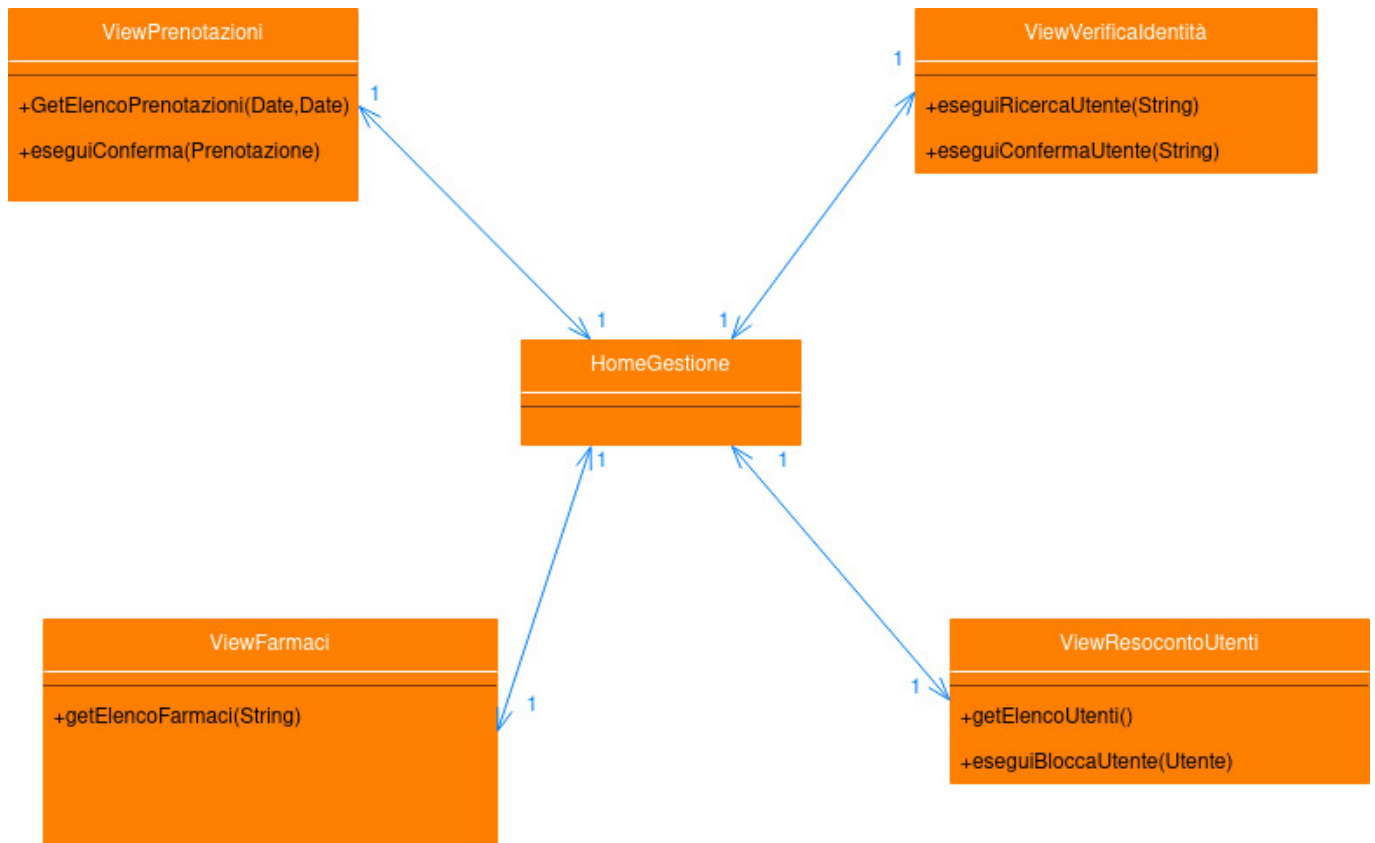


Figura 11: Diagramma di dettaglio: InterfacciaFarmacista

Inserire disegni interfacce qui



Figura 12: Diagramma di dettaglio: InterfacciaGestioneAccesso

L'interfaccia di GestioneAccesso è composta dalle view che permettono a clienti e farmacisti di registrarsi e/o effettuare l'accesso al servizio. Pertanto, è presente in entrambi i client, completata dalle ulteriori interfacce specifiche del client.

Inserire disegni interfacce qui

4.2.2 Interazione

Si riportano di seguito i vari diagrammi di sequenza, aggiornati rispetto a quelli visti in fase di analisi.

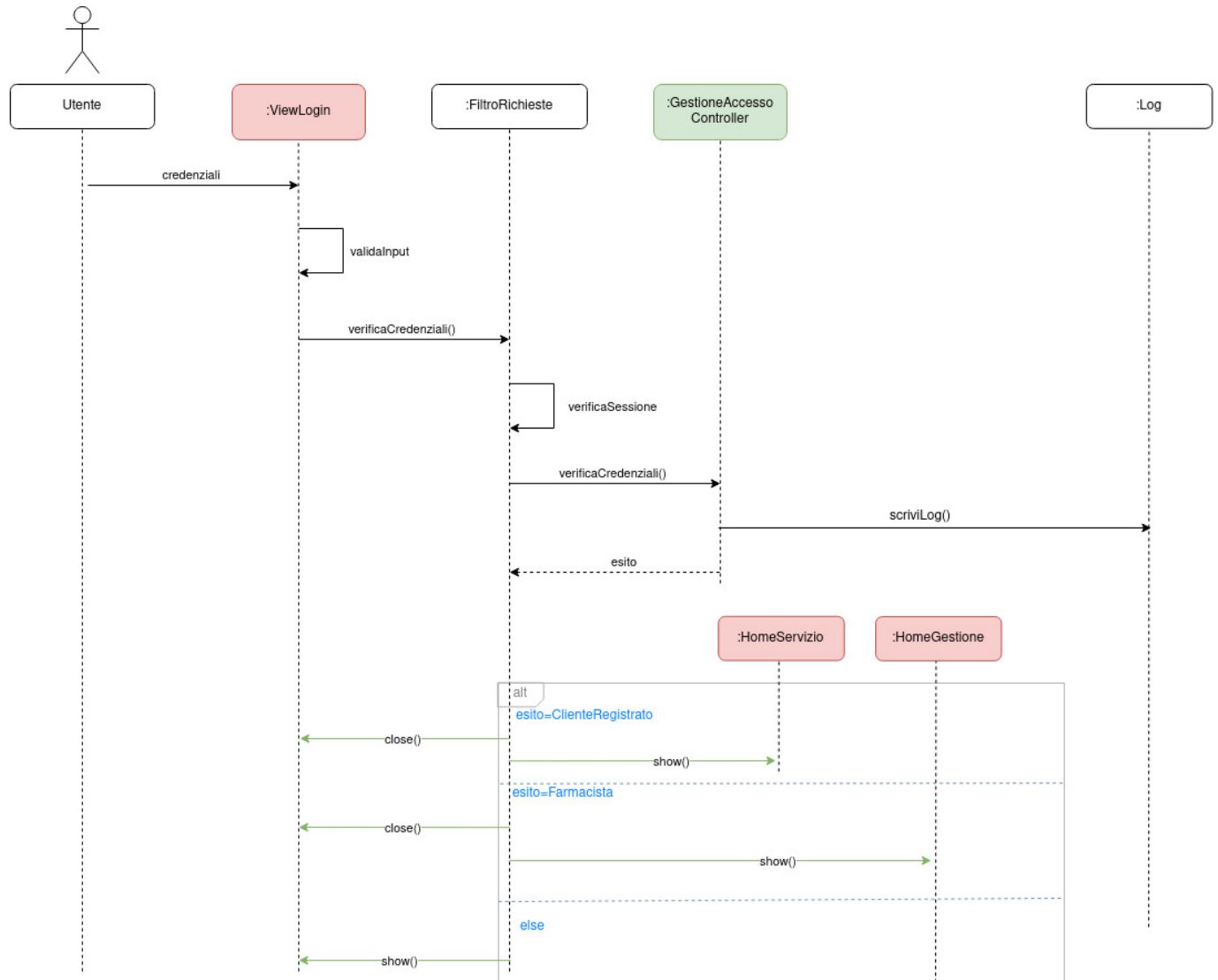


Figura 13: Diagramma di Sequenza: LoginUtente

Registrazione Utente

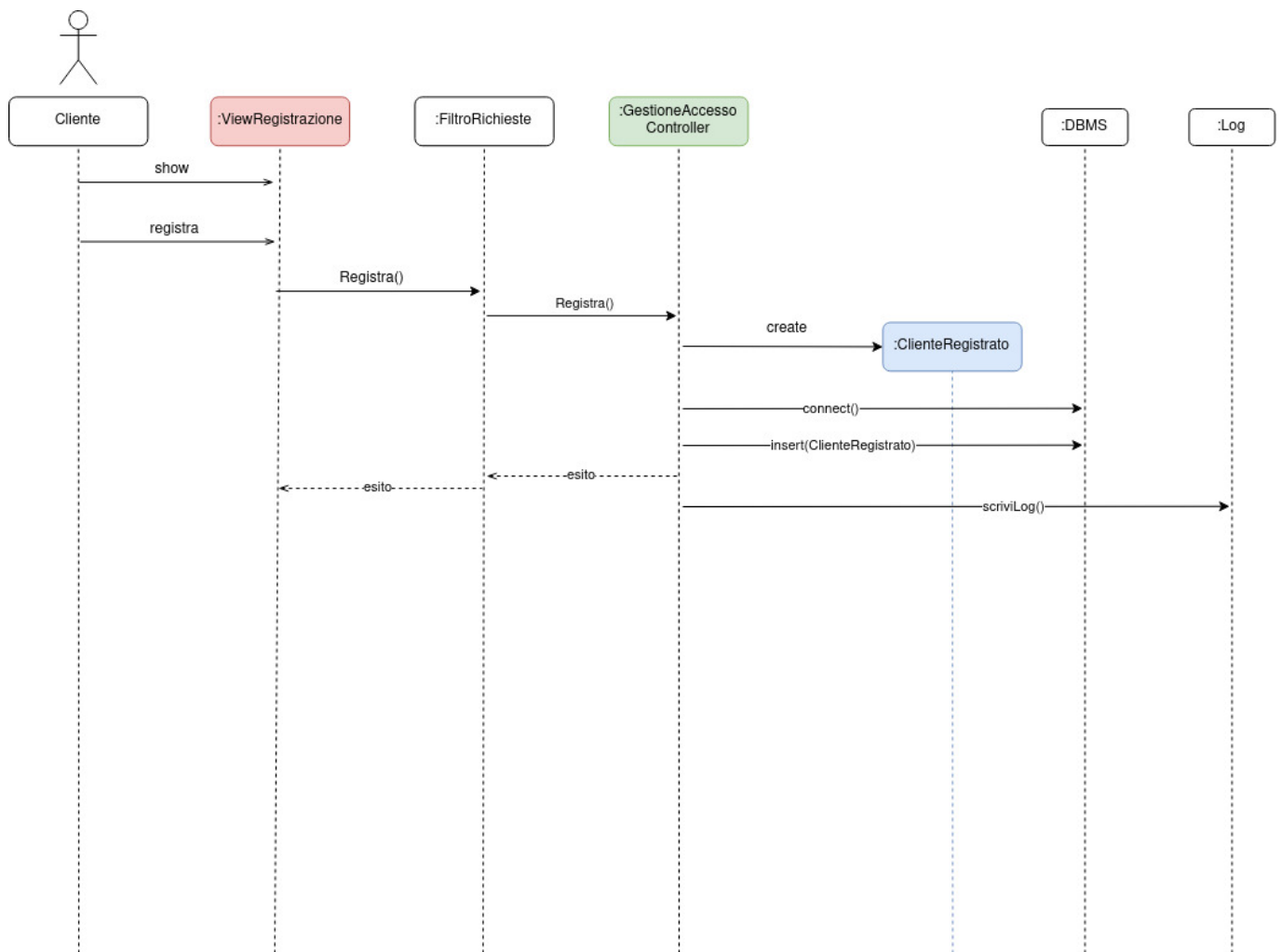
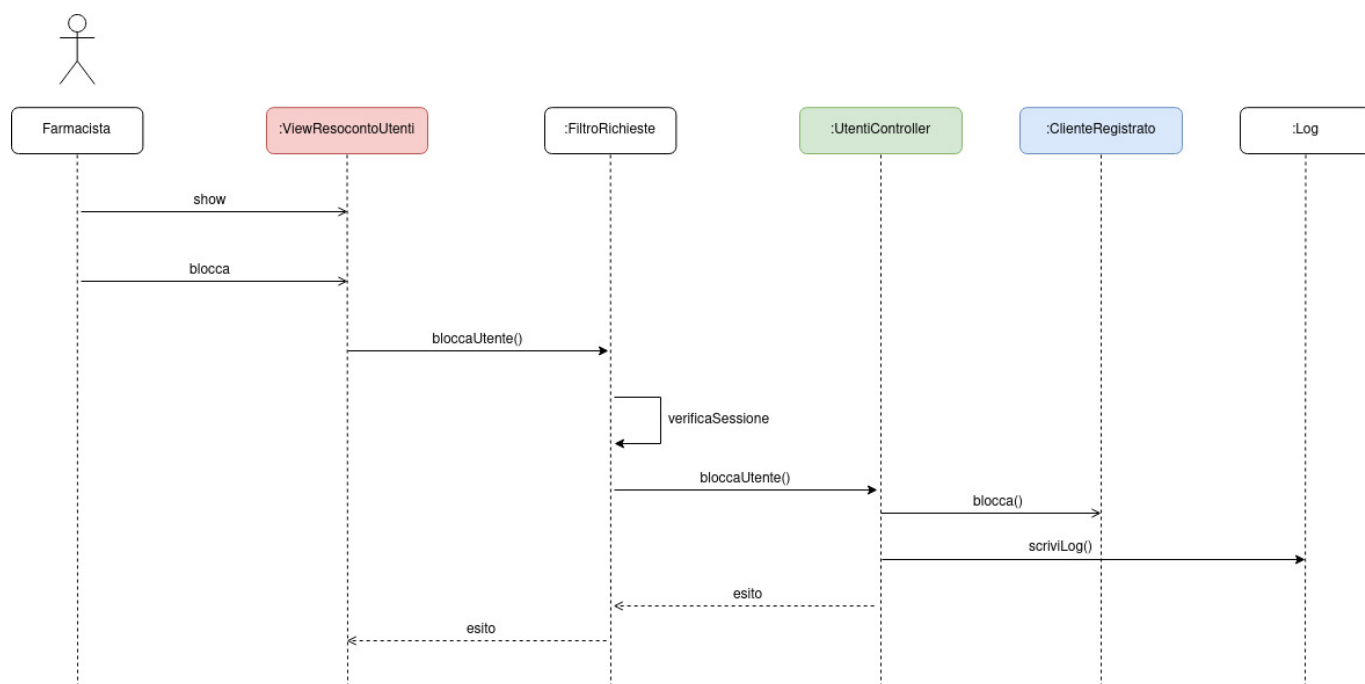
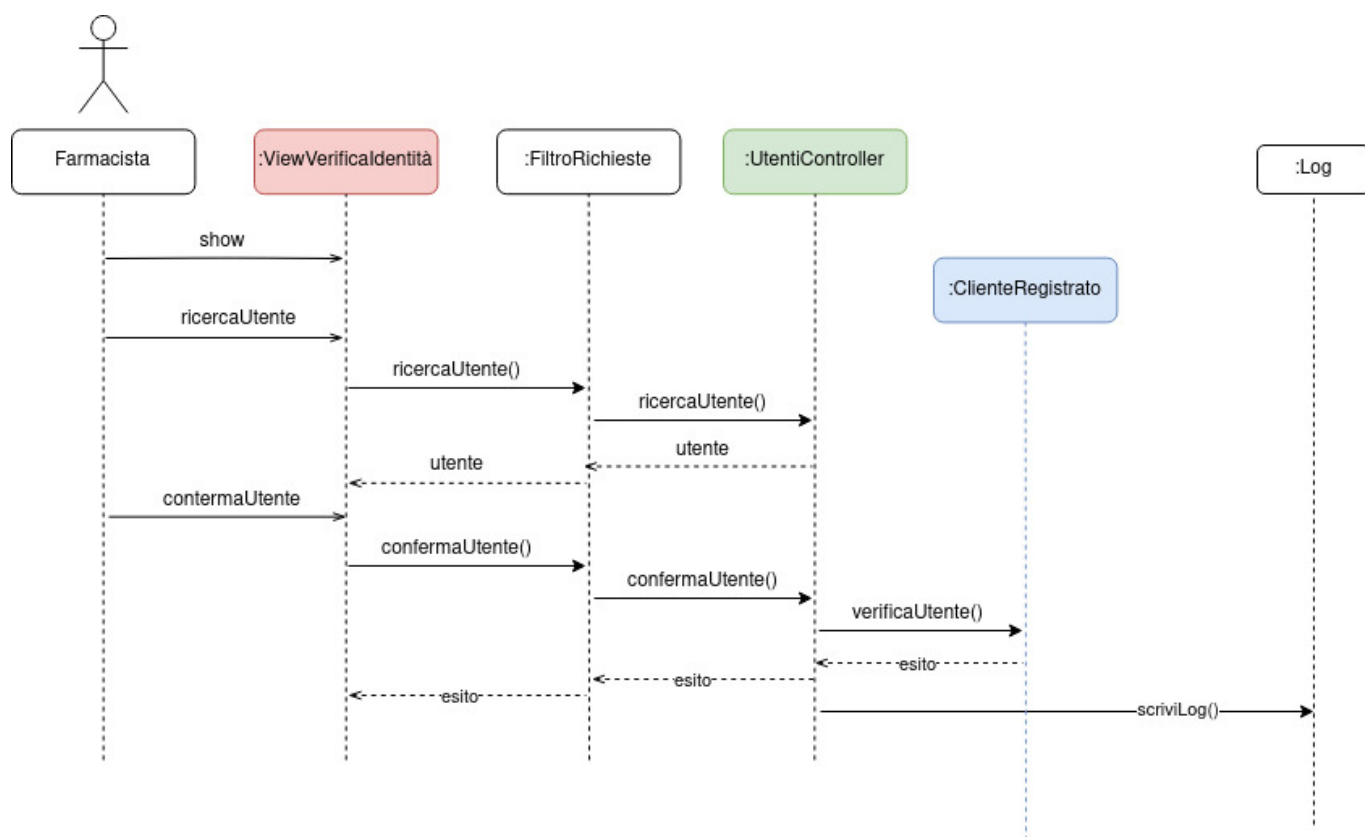


Figura 14: Diagramma di Sequenza: RegistrazioneUtente



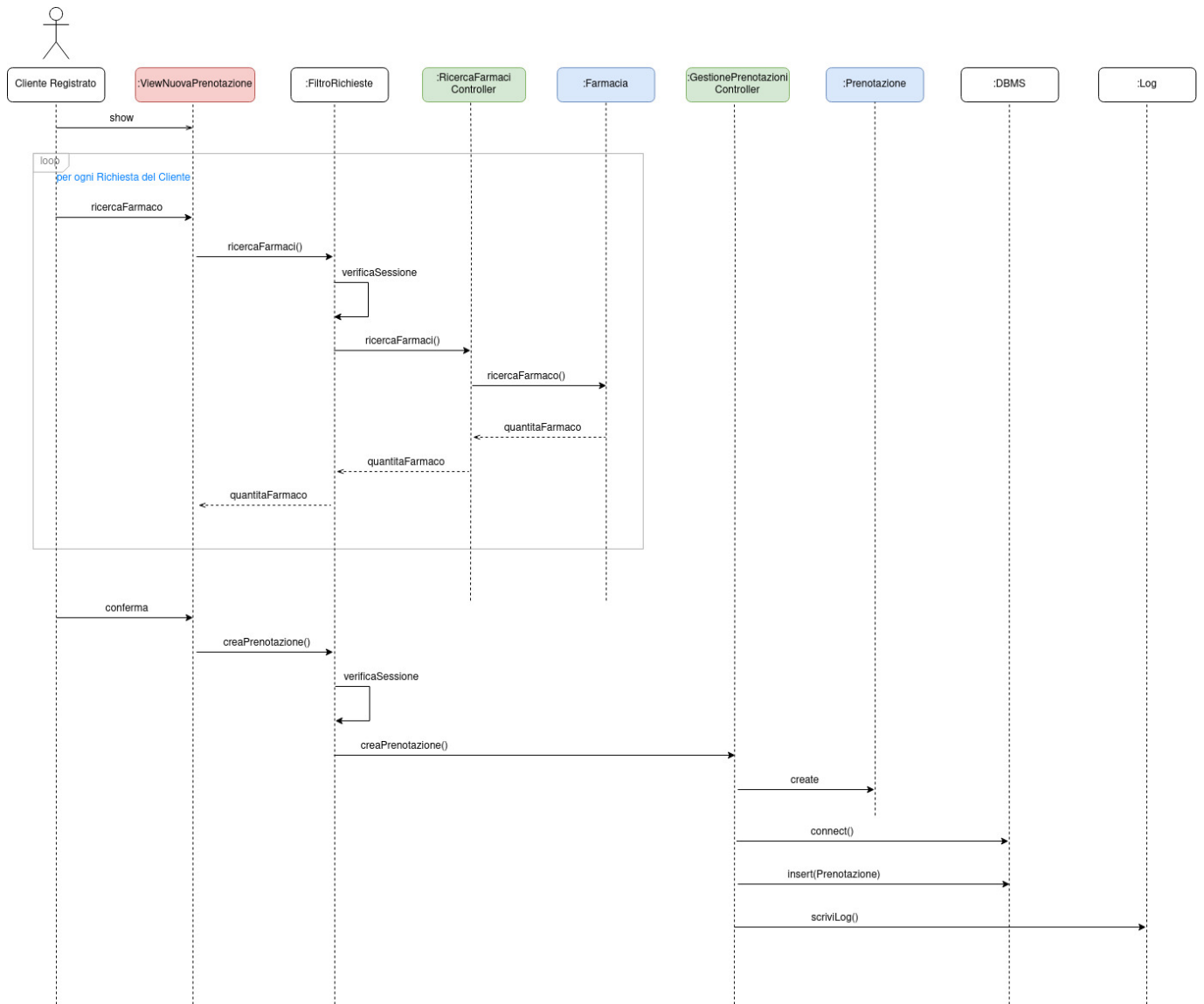


Figura 15: Diagramma di Sequenza: NuovaPrenotazione

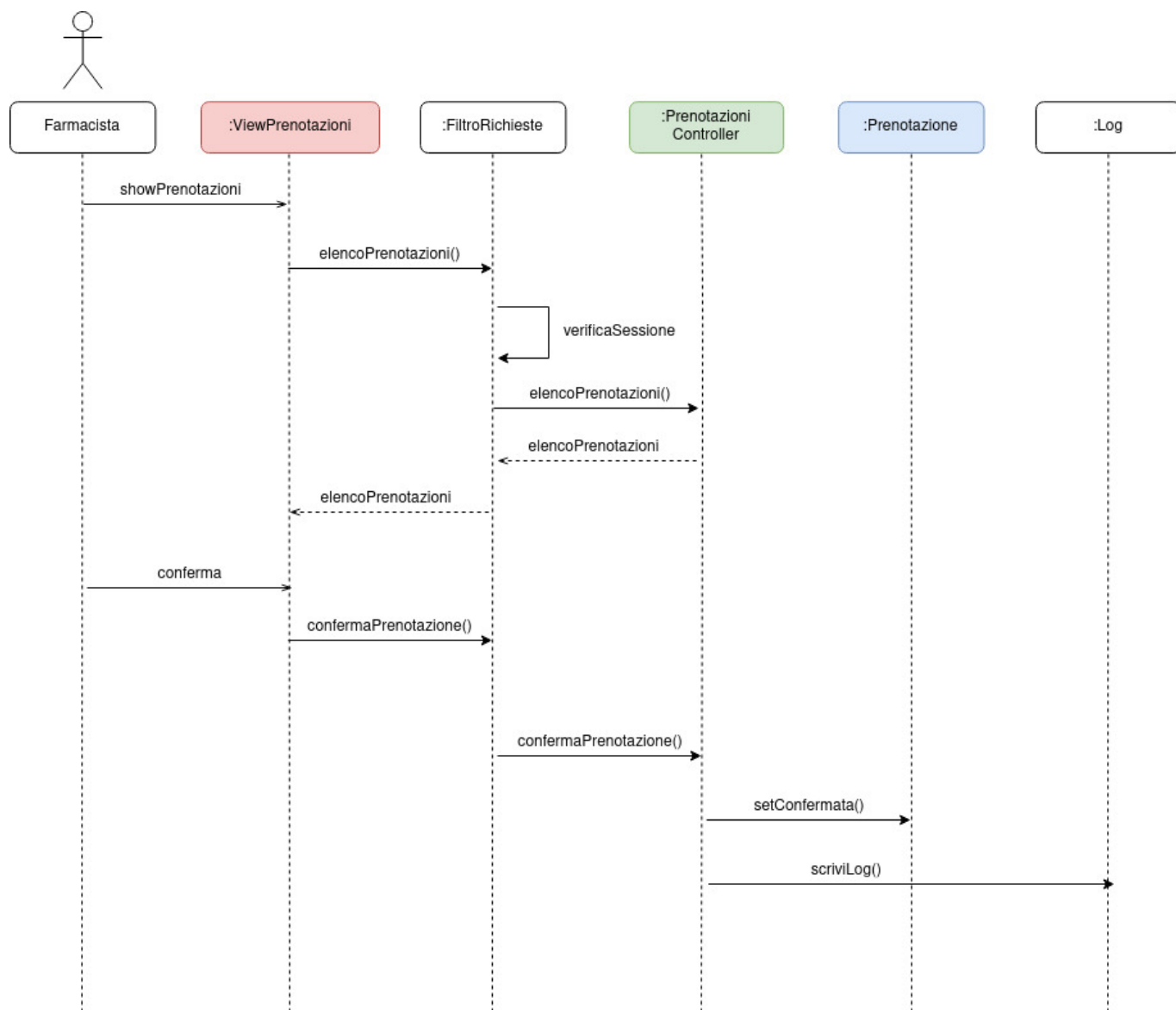


Figura 16: Diagramma di Sequenza: ConfermaPrenotazione

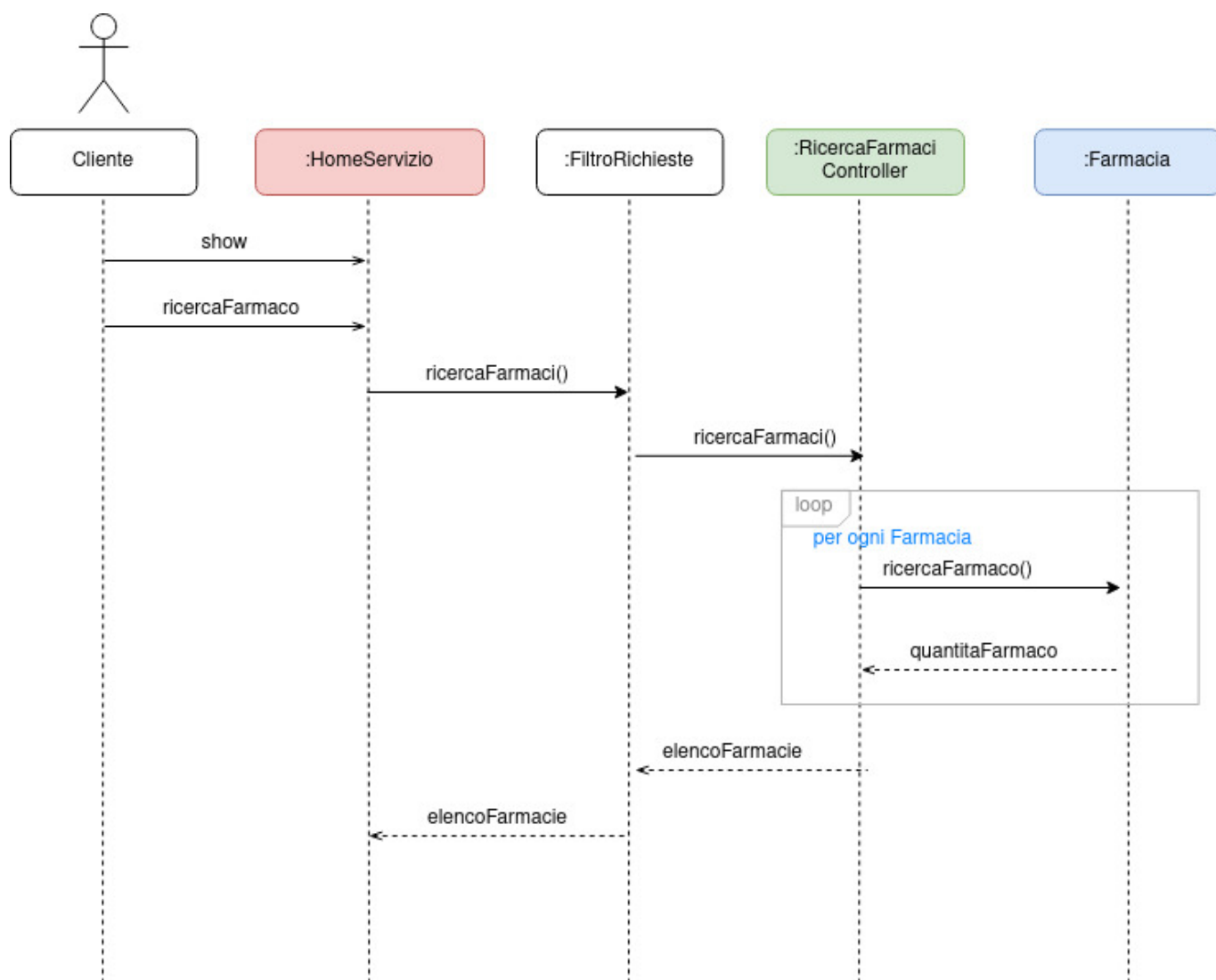
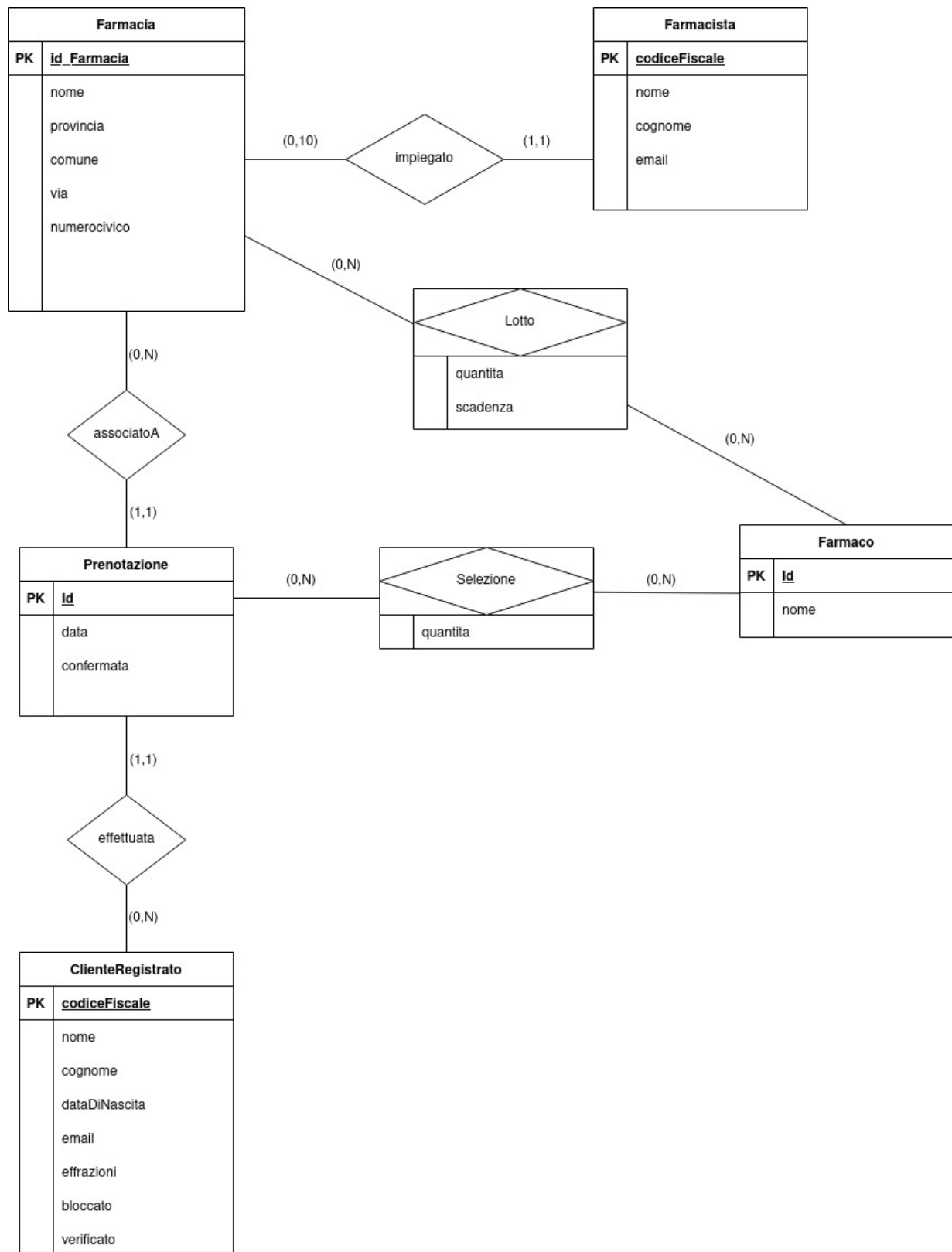


Figura 17: Diagramma di Sequenza: RicercaFarmaci

4.3 Progettazione della persistenza

Diagramma E-R



Come si può notare, il diagramma E-R della persistenza segue precisamente la struttura del modello del dominio mostrato precedentemente. La differenza sta nelle associazioni, che presumibilmente in fase di progettazione logica ed implementazione del database verranno concretizzate in classi di associazione, si avranno quindi due tabelle ulteriori (**Lotto** e **Selezione**) per modellare le associazioni.

4.3.1 Formato dei file di log

Il formato del file di log su cui il sistema terrà traccia delle operazioni sarà il seguente:

Esempio: File `/var/log/farmabyte.log`

`$ Data - Ora - Operazione - Descrizione - ID utente`

Nota: l'ID utente è l'identificativo dell'esecutore di tale operazione, può essere quindi sia un cliente che un farmacista. Se nell'eseguire l'operazione, l'utente interagisce o modifica lo stato di altri utenti questo dovrà essere specificato nella descrizione (Caso tipico: farmacista verifica l'identità di un cliente). L'operazione sarà semplicemente il nome del metodo invocato. Si noti infine che ogni riga del file di log inizia con il carattere `$`, tale carattere viene usato come delimitatore e non sarà quindi utilizzabile all'interno della descrizione (Né nella definizione degli altri campi).

4.4 Progettazione del collaudo

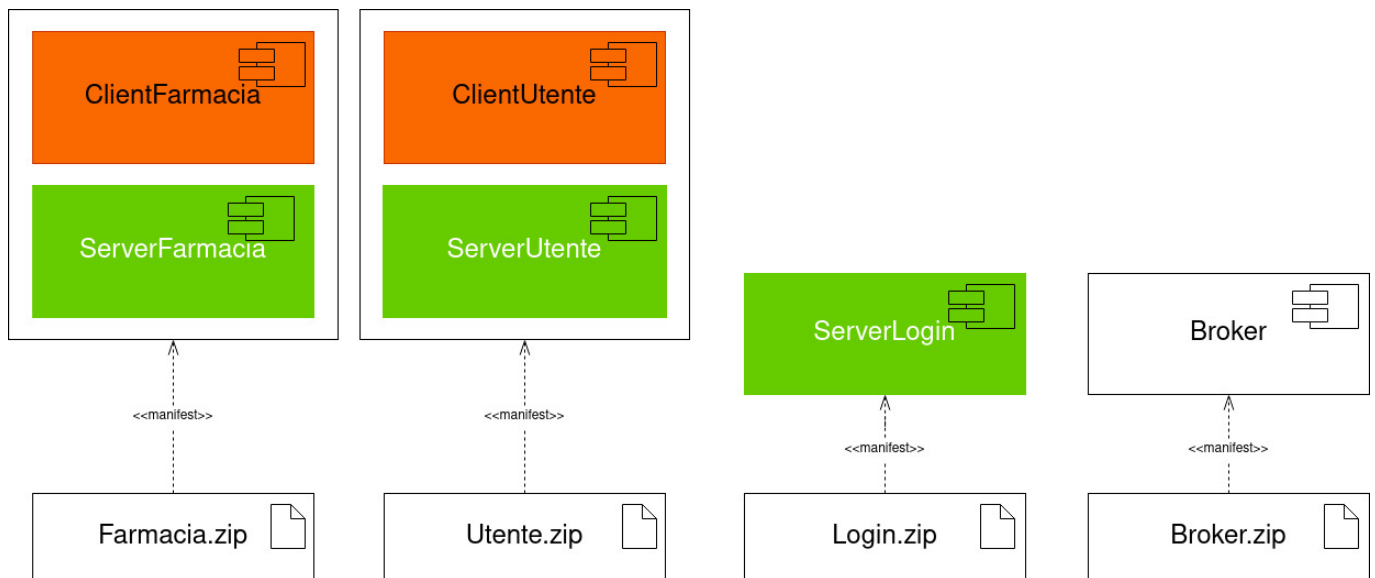
Roba

4.5 Progettazione per il deployment

Poiché si è scelto di realizzare il programma come applicazione web, il deployment non necessiterà di particolari configurazioni lato client. Tutte le interfacce (per clienti, farmacisti, login) sono in realtà memorizzate nel server locale e potranno essere quindi aggiornate automaticamente e in modo centralizzato dagli amministratori di sistema.

4.6 Deployment

4.6.1 Artefatti



4.6.2 Deployment Type-Level

