

UNIVERSITA' DEGLI STUDI DI
NAPOLI FEDERICO II

Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Ingegneria Informatica

Corso di Ingegneria del Software
a.a. 2023/24

Gestione di una farmacia



Professoressa: Anna Rita Fasolino

Autori: Carmine Colucci -
Antonio Capone -
Luigi Auggiero -

Indice

1	Specifiche Informali	5
2	Analisi e Specifica dei Requisiti	6
2.1	Analisi Nomi-Verbi	6
2.2	Revisione dei Requisiti	6
2.2.1	Requisiti Aggiuntivi	7
2.3	Glossario dei Termini	7
2.4	Classificazione dei Requisiti	8
2.4.1	Requisiti Funzionali	8
2.4.2	Requisiti sui Dati	8
2.4.3	Vincoli/Altri Requisiti	8
3	Modellazione dei Casi d'Uso	9
3.1	Attori e Casi d'Uso	9
3.2	Diagramma dei Casi d'Uso	10
3.3	Scenari	11
3.4	Diagramma delle Classi	16
3.5	Diagrammi di Sequenza	16
3.6	Verifica Completezza dei Requisiti	23
4	Piano di Test Funzionale	24
4.1	RegistraCliente	24
4.2	LoginUtente	26
4.3	AggiungiFarmaco	27
4.4	ModificaFarmaco	28
4.5	CercaFarmaco	29
4.6	EliminaFarmaco	29
4.7	GeneraReport	30
4.8	GeneraOrdineAcquistoFarmacista	30
4.9	RegistraConsegnaOrdineAcquisto	31
4.10	RitiraOrdine	31
4.11	CreaOrdine	32
5	Progettazione	33
5.1	Diagramma delle classi	33
5.2	Modello Entity-Relationship del Database	34
5.3	Diagrammi di sequenza	34
6	Implementazione	35
6.1	Specifiche di implementazione Java	35
6.2	Artefatti necessari per l'installazione e l'esecuzione	37
6.3	Documentazione	37
6.4	Diagramma di deployment	38
7	Testing	39
7.1	Test strutturale	39
7.1.1	creaOrdine()	39
7.1.2	modificaFarmaco()	42
7.2	Test funzionale	43
7.2.1	RegistraCliente	43

7.2.2	LoginUtente	44
7.2.3	AggiungiFarmaco	44
7.2.4	CercaFarmaco	45
7.2.5	EliminaFarmaco	45
7.2.6	ModificaFarmaco	45
7.2.7	GeneraReport	45
7.2.8	GeneraOrdineAcquistoFarmacista	46
7.2.9	RitiraOrdine	46
7.2.10	RegistraConsegnaOrdineAcquisto	46
7.2.11	CreaOrdine	47

Capitolo 1

Specifiche Informali

Il progetto consiste nello sviluppo di un sistema integrato per la gestione di una farmacia, che incorpora funzionalità avanzate per ottimizzare sia le operazioni interne che le interazioni con i clienti.

Il sistema gestisce le vendite di farmaci online ed al tal fine possiede un catalogo di tutti i farmaci che sono in vendita. Il catalogo dunque contiene un insieme di farmaci, caratterizzati da nome, codice identificativo (stringa alfa-numerica di 20 caratteri), prezzo e tipologia di farmaco: farmaco da banco o farmaco da prescrizione. Il sistema deve permettere al farmacista di poter visualizzare il catalogo e di aggiornarlo, potendo aggiungere, modificare o eliminare un prodotto.

I clienti possono accedere al catalogo online della farmacia solo previa registrazione, ed un database mantiene salvate le loro informazioni personali, incluso lo storico dei farmaci da essi acquistati, sia normalmente che tramite prescrizione.

Il sistema deve presentare un'interfaccia al cliente che permette di creare un ordine consultando il catalogo, e di poter selezionare l'eventuale possesso della prescrizione per poter acquistare i farmaci non da banco. Nel caso in cui un utente provi ad acquistare un farmaco senza prescrizione il sistema deve generare un opportuno messaggio di errore.

Ogni volta che un cliente effettua un ordine le quantità in magazzino del farmaco devono essere opportunamente decrementate ed in caso di terminazione delle scorte, un ordine di acquisto deve essere creato con una quantità di prodotto da ordinare di default. È compito del farmacista poter visualizzare l'elenco degli ordini d'acquisto in corso e registrare l'avvenuta consegna di un ordine, con opportuno incremento delle quantità in magazzino.

Infine, una caratteristica distintiva di questo sistema è il ruolo del "Direttore della Farmacia", che ha accesso a strumenti analitici e di reporting per una gestione strategica delle attività commerciali. Il direttore ha la capacità di generare report dettagliati. Questi report forniscono insight sui farmaci venduti e sull'incasso generato in un dato periodo (in una specifica data oppure in un intervallo compreso fra due date). In particolare, il sistema differenzia tra le vendite da banco, che contribuiscono all'incasso della farmacia, e le vendite di farmaci su prescrizione, che non generano incassi diretti. I report includeranno dati come il numero totale di farmaci venduti, la categoria di vendita (da banco o su prescrizione) e l'incasso totale delle vendite da banco.

Capitolo 2

Analisi e Specifica dei Requisiti

2.1 Analisi Nomi–Verbi

Il progetto consiste nello sviluppo di un sistema integrato per la gestione di una farmacia, che incorpora funzionalità avanzate per ottimizzare sia le operazioni interne che le interazioni con i clienti.

Il sistema gestisce le vendite di farmaci online ed al tal fine possiede un **catalogo** di tutti i farmaci che sono in vendita. Il catalogo dunque contiene un insieme di **farmaci**, caratterizzati da **nome**, **codice identificativo** (stringa alfa-numerica di 20 caratteri), **prezzo** e **tipologia di farmaco**: farmaco da banco o farmaco da prescrizione. Il sistema deve permettere al farmacista di poter **visualizzare il catalogo** e di aggiornarlo, potendo **aggiungere, modificare o eliminare un prodotto**.

I **clienti** possono **accedere al catalogo** online della farmacia solo previa registrazione, ed un database mantiene salvate le loro **informazioni personali**, incluso lo **storico dei farmaci da essi acquistati**, sia normalmente che tramite prescrizione.

Il sistema deve presentare un'interfaccia al cliente che permette di **creare un ordine consultando il catalogo**, e di poter selezionare l'eventuale possesso della prescrizione per poter acquistare i farmaci non da banco. **Nel caso in cui un utente provi ad acquistare un farmaco senza prescrizione il sistema deve generare un opportuno messaggio di errore.**

Ogni volta che un cliente effettua un ordine le quantità in magazzino del farmaco devono essere opportunamente decrementate ed in caso di terminazione delle scorte, un **ordine di acquisto** deve essere creato con una quantità di prodotto da ordinare di default. È compito del **farmacista** poter **visualizzare l'elenco degli ordini d'acquisto in corso** e registrare l'avvenuta consegna di un ordine, con opportuno incremento delle quantità in magazzino.

Infine, una caratteristica distintiva di questo sistema è il ruolo del “**Direttore della Farmacia**”, che ha accesso a strumenti analitici e di reporting per una gestione strategica delle attività commerciali. Il direttore ha la capacità di **generare report dettagliati**. Questi **report** forniscono insight sui **farmaci venduti** e sull'**incasso generato in un dato periodo** (in una specifica data oppure in un intervallo compreso fra due date). In particolare, il sistema differenzia tra le vendite da banco, che contribuiscono all'incasso della farmacia, e le vendite di farmaci su prescrizione, che non generano incassi diretti. I report includeranno dati come **il numero totale di farmaci venduti**, **la categoria di vendita** (da banco o su prescrizione) e **l'incasso totale delle vendite da banco**.

◆ Classe	◆ Attributo	◆ Funzionalità	◆ Attore	◆ Classe–Attore
----------	-------------	----------------	----------	-----------------

2.2 Revisione dei Requisiti

1. I farmaci sono caratterizzati da nome, codice identificativo, prezzo e tipologia di farmaco;
2. Un farmaco può essere da banco o da prescrizione;
3. Il sistema deve permettere al farmacista di visualizzare il catalogo;
4. Il sistema deve permettere al farmacista di aggiungere un prodotto;
5. Il sistema deve permettere al farmacista di modificare un prodotto;
6. Il sistema deve permettere al farmacista di eliminare un prodotto;
7. Il sistema deve permettere al cliente di registrarsi;

8. Di ogni cliente si vogliono memorizzare le informazioni personali e lo storico dei farmaci acquistati;
9. Il sistema deve permettere al cliente di visualizzare il catalogo;
10. Il sistema deve permettere al cliente di creare un ordine consultando il catalogo;
11. Il sistema deve generare un messaggio di errore se il cliente prova ad acquistare un farmaco da prescrizione senza averla;
12. Ogni volta che un cliente effettua un ordine, il sistema deve opportunamente aggiornare le scorte;
13. Il sistema deve generare un ordine di acquisto in caso di esaurimento delle scorte di un prodotto;
14. Il sistema deve permettere al farmacista di visualizzare l'elenco degli ordini d'acquisto in corso;
15. Il sistema deve permettere al farmacista di registrare l'avvenuta consegna di un ordine d'acquisto, con opportuno incremento delle quantità in magazzino;
16. Il sistema deve permettere al Direttore di generare report dettagliati;
17. Il report fornisce informazioni sul numero totale di farmaci venduti, sulla loro categoria di vendita e sull'incasso totale in un dato periodo;
18. Nella generazione dei report, soltanto le vendite da banco contribuiscono al calcolo dell'incasso totale;

2.2.1 Requisiti Aggiuntivi

In seguito ad un ulteriore colloquio tenuto con il committente, si è ritenuto opportuno introdurre i seguenti requisiti:

19. Il sistema deve permettere al farmacista di visualizzare gli ordini dei clienti;
20. Il sistema deve permettere al farmacista di gestire gli ordini dei clienti;
21. Il sistema deve permettere al cliente di visualizzare i propri ordini;
22. Il sistema deve generare una ricevuta per il cliente in seguito alla creazione di un ordine.
23. Il sistema deve permettere al cliente di cercare un farmaco all'interno del catalogo.
24. Il sistema deve permettere al farmacista di ordinare nuove forniture di un farmaco.

2.3 Glossario dei Termini

Termine	Descrizione	Sinonimi
Farmaco	Un articolo in vendita nella farmacia	Prodotto
Ordine	Ordine generato dal cliente	
Ordine d'acquisto	Ordine di rifornimento generato dal sistema o effettuato dal farmacista	
Report	Documento che fornisce al Direttore insights sui farmaci venduti e sull'incasso generato in un dato periodo	

2.4 Classificazione dei Requisiti

2.4.1 Requisiti Funzionali

ID	Requisito	Origine
RF ₀₁	Il sistema deve permettere al farmacista di visualizzare il catalogo	3
RF ₀₂	Il sistema deve permettere al farmacista di aggiungere un prodotto	4
RF ₀₃	Il sistema deve permettere al farmacista di modificare un prodotto	5
RF ₀₄	Il sistema deve permettere al farmacista di eliminare un prodotto	6
RF ₀₅	Il sistema deve permettere al cliente di registrarsi	7
RF ₀₆	Il sistema deve permettere al cliente di visualizzare il catalogo	9
RF ₀₇	Il sistema deve permettere al cliente di creare un ordine consultando il catalogo	10
RF ₀₈	Il sistema deve generare un messaggio di errore se il cliente prova ad acquistare un farmaco da prescrizione senza averla	11
RF ₀₉	Ogni volta che un cliente effettua un ordine, il sistema deve opportunamente aggiornare le scorte	12
RF ₁₀	Il sistema deve generare un ordine di acquisto in caso di esaurimento delle scorte di un prodotto	13
RF ₁₁	Il sistema deve permettere al farmacista di visualizzare l'elenco degli ordini d'acquisto in corso	14
RF ₁₂	Il sistema deve permettere al farmacista di registrare l'avvenuta consegna di un ordine d'acquisto, con opportuno incremento delle quantità in magazzino	15
RF ₁₃	Il sistema deve permettere al Direttore di generare report dettagliati	16
RF ₁₄	Il sistema deve permettere al farmacista di visualizzare gli ordini dei clienti	19
RF ₁₅	Il sistema deve permettere al farmacista di gestire gli ordini dei clienti	20
RF ₁₆	Il sistema deve permettere al cliente di visualizzare i propri ordini	21
RF ₁₇	Il sistema deve generare una ricevuta per il cliente in seguito alla creazione di un ordine	22
RF ₁₈	Il sistema deve permettere al cliente di cercare un farmaco all'interno del catalogo	23
RF ₁₉	Il sistema deve permettere al farmacista di ordinare nuove forniture di un farmaco.	24

2.4.2 Requisiti sui Dati

ID	Requisito	Origine
RD ₀₁	I farmaci sono caratterizzati da nome, codice identificativo, prezzo e tipologia di farmaco	1
RD ₀₂	Un farmaco può essere da banco o da prescrizione	2
RD ₀₃	Di ogni cliente si vogliono memorizzare le informazioni personali e lo storico dei farmaci acquistati	8
RD ₀₄	Il report fornisce informazioni sul numero totale di farmaci venduti, sulla loro categoria di vendita e sull'incasso totale in un dato periodo	17

2.4.3 Vincoli/Altri Requisiti

ID	Requisito	Origine
V ₀₁	Nella generazione dei report, soltanto le vendite da banco contribuiscono al calcolo dell'incasso totale	18

Capitolo 3

Modellazione dei Casi d'Uso

3.1 Attori e Casi d'Uso

Attori primari

- UtenteRegistrato
- Cliente
- Farmacista
- Direttore
- Cliente non registrato

Attori secondari

- Farmacista

Casi d'uso

1. VisualizzaCatalogo
2. AggiungiFarmaco
3. CercaFarmaco
4. ModificaFarmaco
5. EliminaFarmaco
6. RegistraCliente
7. CreaOrdine
8. GeneraOrdineAcquisto
9. VisualizzaOrdiniAcquisto
10. RegistraConsegnaOrdineAcquisto
11. GeneraReport
12. VisualizzaOrdiniFarmacia
13. RitiraOrdine
14. VisualizzaStoricoOrdini
15. GeneraOrdineAcquistoFarmacista

Casi d'uso di inclusione

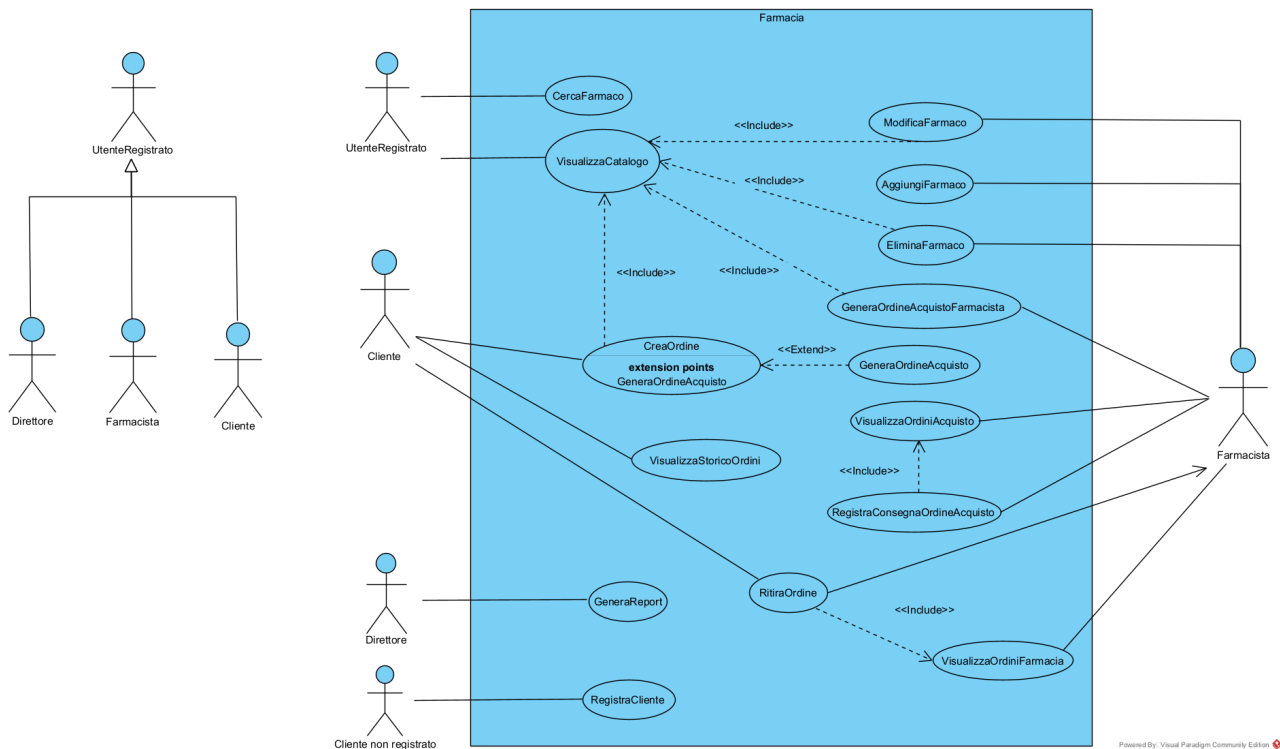
1. CercaFarmaco
2. VisualizzaOrdiniFarmacia
3. VisualizzaOrdiniAcquisto

Casi d'uso di estensione

1. GeneraOrdineAcquisto

Caso d'uso	Attori primari	Attori secondari	Incl. / Ext.	Requisito
VisualizzaCatalogo	UtenteRegistrato	–	–	RF01 RF06
AggiungiFarmaco	Farmacista	–	–	RF02
ModificaFarmaco	Farmacista	–	Include VisualizzaCatalogo	RF03
EliminaFarmaco	Farmacista	–	Include VisualizzaCatalogo	RF04
RegistraCliente	Cliente non registrato	–	–	RF05
CreaOrdine	Cliente	–	Include VisualizzaCatalogo	RF07 RF08 RF09 RF17
GeneraOrdineAcquisto	Cliente	–	Estende CreaOrdine	RF10
VisualizzaOrdiniAcquisto	Farmacista	–	–	RF11
RegistraConsegnaOrdineAcquisto	Farmacista	–	Include VisualizzaOrdiniAcquisto	RF12
GeneraReport	Direttore	–	–	RF13
VisualizzaOrdiniFarmacia	Farmacista	–	–	RF14
RitiraOrdine	Cliente	Farmacista	Include VisualizzaOrdiniFarmacia	RF15
VisualizzaStoricoOrdini	Cliente	–	–	RF16
CercaFarmaco	UtenteRegistrato	–	–	RF18
GeneraOrdineAcquistoFarmacista	Farmacista	–	Include VisualizzaCatalogo	RF19

3.2 Diagramma dei Casi d'Uso



3.3 Scenari

Caso d'uso:	RegistraCliente
Attore primario	Cliente non registrato
Attore secondario	Nessuno
Descrizione	Un cliente si registra al sistema
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il cliente richiede al sistema di registrarlo 2. Il cliente inserisce le informazioni richieste: username, password, nome, cognome, data di nascita, email (vedi RD₀₃) 3. Il sistema valida il contenuto dei dati inseriti 4. Se il controllo ha esito positivo <ol style="list-style-type: none"> 4.1. Il sistema registra il cliente 5. Altrimenti <ol style="list-style-type: none"> 5.1. Il sistema mostra un messaggio di errore
Post-condizioni	Il cliente viene correttamente registrato nel sistema
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	GeneraReport
Attore primario	Direttore
Attore secondario	–
Descrizione	Il Direttore della farmacia genera un report
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il Direttore richiede al sistema di generare un report 2. Il Direttore inserisce un intervallo di date 3. Il sistema genera e mostra il report
Post-condizioni	Il report viene generato
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	ModificaFarmaco
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista modifica un farmaco presente nel catalogo della farmacia
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di modificare un farmaco nel catalogo 2. <i>include</i> (VisualizzaCatalogo) 3. Il farmacista seleziona il farmaco di interesse 4. Il farmacista inserisce i nuovi dati del farmaco
Post-condizioni	Il sistema aggiorna i dati del farmaco
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso: AggiungiFarmaco	
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista aggiunge un nuovo farmaco al catalogo della farmacia
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di inserire un nuovo farmaco nel catalogo 2. Il farmacista inserisce i dati del farmaco (vedi RD₀₁, RD₀₂) 3. Il sistema valida il contenuto dei dati inseriti 4. Se il controllo ha successo <ol style="list-style-type: none"> 4.1. Il sistema aggiunge il farmaco al catalogo 5. Altrimenti <ol style="list-style-type: none"> 5.1. Il sistema mostra un messaggio di errore
Post-condizioni	Il sistema aggiunge il farmaco al catalogo
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso: EliminaFarmaco	
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista elimina un farmaco presente nel catalogo della farmacia
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di eliminare un farmaco nel catalogo 2. <i>include</i> (VisualizzaCatalogo) 3. Il farmacista seleziona il farmaco di interesse 4. Il farmacista conferma la rimozione del farmaco
Post-condizioni	Il sistema elimina il farmaco dal catalogo
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso: CercaFarmaco	
Attore primario	UtenteRegistrato
Attore secondario	Nessuno
Descrizione	L'utente cerca un farmaco all'interno del catalogo
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'utente richiede al sistema di cercare un farmaco nel catalogo 2. L'utente inserisce il nome del farmaco 3. Se il farmaco è presente nel catalogo <ol style="list-style-type: none"> 3.1. Il sistema mostra le informazioni del farmaco 4. Altrimenti <ol style="list-style-type: none"> 4.1. Il sistema mostra un messaggio di errore
Post-condizioni	–
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	VisualizzaCatalogo
Attore primario	UtenteRegistrato
Attore secondario	Nessuno
Descrizione	Un utente visualizza il catalogo della farmacia
Pre-condizioni	Aver effettuato l'accesso al sistema
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. L'utente richiede al sistema di visualizzare il catalogo 2. Il sistema mostra il catalogo della farmacia
Post-condizioni	–
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	VisualizzaOrdiniAcquisto
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista visualizza l'elenco degli ordini d'acquisto della farmacia
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di visualizzare gli ordini d'acquisto 2. Il sistema mostra l'elenco degli ordini d'acquisto
Post-condizioni	
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	RegistraConsegnaOrdineAcquisto
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista registra l'avvenuta consegna di un ordine d'acquisto
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di visualizzare gli ordini d'acquisto 2. <i>include</i> (VisualizzaOrdiniAcquisto) 3. Il farmacista registra l'avvenuta consegna della fornitura di interesse
Post-condizioni	La consegna dell'ordine di acquisto è registrata e le scorte sono aggiornate
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	VisualizzaOrdiniFarmacia
Attore primario	Farmacista
Attore secondario	Nessuno
Descrizione	Un farmacista visualizza l'elenco degli ordini effettuati dai clienti
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il farmacista richiede al sistema di visualizzare gli ordini dei clienti 2. Il sistema mostra l'elenco degli ordini dei clienti
Post-condizioni	
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	VisualizzaStoricoOrdini
Attore primario	Cliente
Attore secondario	–
Descrizione	Un cliente visualizza lo storico dei suoi ordini
Pre-condizioni	Il cliente è autenticato
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il cliente richiede al sistema di visualizzare lo storico dei suoi ordini 2. Il sistema mostra lo storico degli ordini del cliente
Post-condizioni	
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	RitiraOrdine
Attore primario	Cliente
Attore secondario	Farmacista
Descrizione	Un cliente ritira l'ordine effettuato online
Pre-condizioni	Esiste l'ordine nel sistema
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il cliente arriva in farmacia e richiede il ritiro di un ordine mostrando la ricevuta 2. <i>include</i> (VisualizzaOrdiniFarmacia) 3. Il farmacista seleziona l'ordine di interesse 4. Il farmacista richiede al sistema di modificare lo stato dell'ordine
Post-condizioni	Lo stato dell'ordine è stato correttamente modificato
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	–

Caso d'uso:	GeneraOrdineAcquistoFarmacista
Attore primario	Farmacista
Attore secondario	–
Descrizione	Il farmacista genera un ordine di fornitura
Pre-condizioni	–
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Per ogni farmaco da ordinare <ol style="list-style-type: none"> 1.1. <i>include</i> (CercaFarmaco) 1.2. Il farmacista inserisce la quantità da ordinare 1.3. Il sistema aggiunge il farmaco all'ordine d'acquisto 2. Il sistema crea l'ordine d'acquisto
Post-condizioni	L'ordine di acquisto viene generato con successo
Casi d'uso correlati	–
Sequenza di eventi alternativa	–

Caso d'uso:	CreaOrdine
Attore primario	Cliente
Attore secondario	–
Descrizione	Un cliente genera un ordine contenente farmaci presenti in catalogo
Pre-condizioni	Il cliente è autenticato
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Il cliente richiede al sistema di creare un ordine 2. include (VisualizzaCatalogo) 3. Finché il cliente vuole aggiungere farmaci all'ordine <ol style="list-style-type: none"> 3.1. Se viene richiesto un farmaco non da banco, il sistema chiede conferma del possesso della prescrizione 3.2. Il cliente seleziona la quantità di farmaco desiderata 3.3. Il sistema verifica che le scorte siano sufficienti 3.4. Il sistema crea l'ordine con i farmaci inseriti e aggiorna le scorte 3.5. <i>punto di estensione</i>: GeneraOrdineAcquisto 4. Il sistema mostra al cliente il resoconto dell'ordine e l'importo totale 5. Se il cliente conferma l'acquisto <ol style="list-style-type: none"> 5.1. Il sistema restituisce una ricevuta al cliente 6. Altrimenti <ol style="list-style-type: none"> 6.1. Il sistema elimina l'ordine
Post-condizioni	L'ordine viene correttamente generato
Casi d'uso correlati	Nessuno
Sequenza di eventi alternativa	<ol style="list-style-type: none"> 1. Al punto 3.1, il sistema mostra un messaggio di errore se il cliente non è in possesso della prescrizione <ol style="list-style-type: none"> 1.1. L'ordine non viene creato 2. Al punto 3.3, il sistema rileva che le scorte non sono sufficienti <ol style="list-style-type: none"> 2.1. L'ordine non viene creato

Caso d'uso d'estensione:	GeneraOrdineAcquisto
Attore primario	Cliente
Attore secondario	–
Descrizione	Il sistema genera un ordine di fornitura
Pre-condizioni	Le scorte di uno o più prodotti terminano in seguito ad un ordine
Sequenza di eventi principale	<ol style="list-style-type: none"> 1. Per ogni farmaco terminato <ol style="list-style-type: none"> 1.1. Il sistema aggiunge il farmaco all'ordine d'acquisto con una quantità di default 2. Il sistema crea l'ordine d'acquisto
Post-condizioni	L'ordine di acquisto viene generato con successo
Casi d'uso correlati	CreaOrdine
Sequenza di eventi alternativa	–

3.4 Diagramma delle Classi

Di seguito riportiamo il diagramma delle classi di analisi.

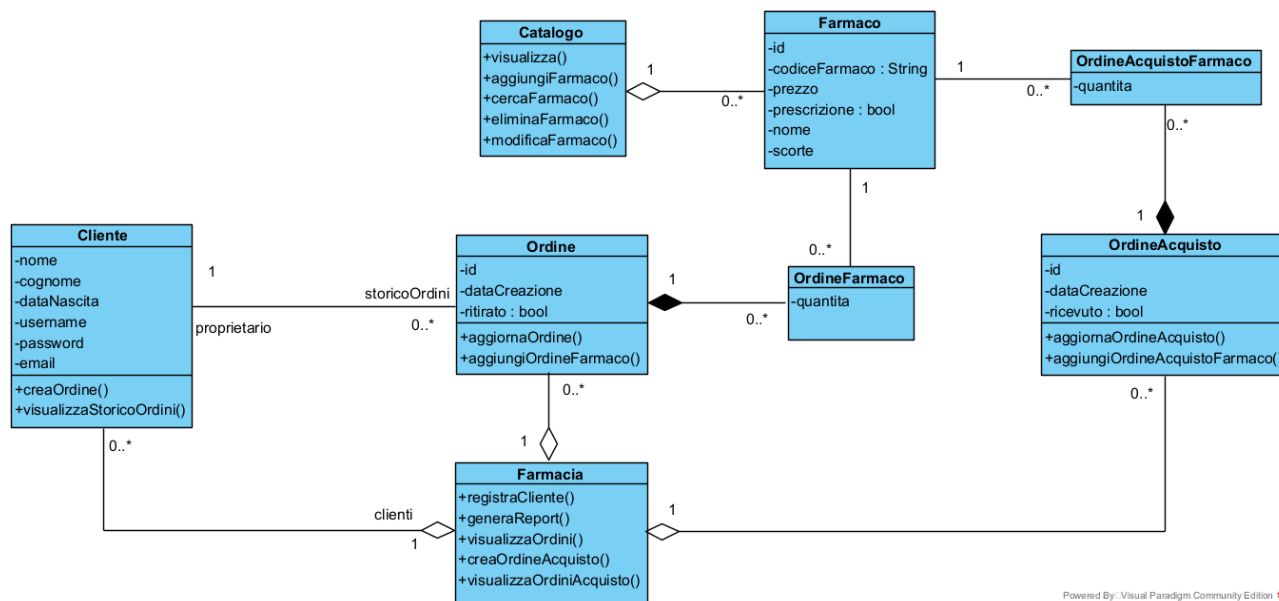


Figura 3.1: Diagramma delle classi di analisi

3.5 Diagrammi di Sequenza

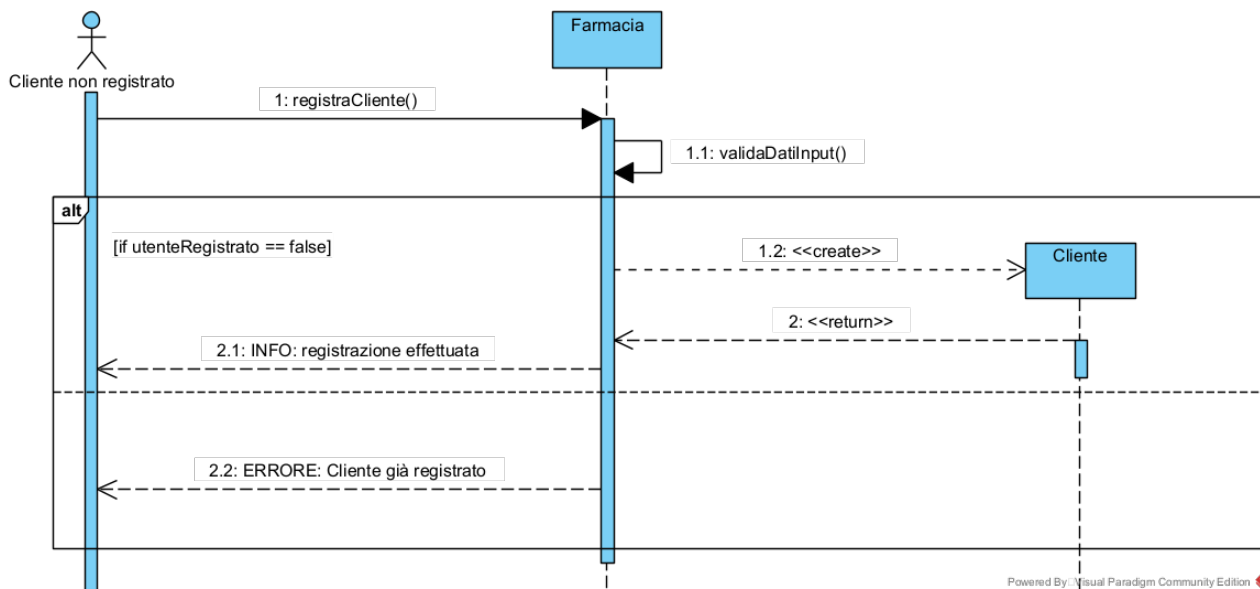


Figura 3.2: Diagramma di sequenza di analisi di RegistraCliente

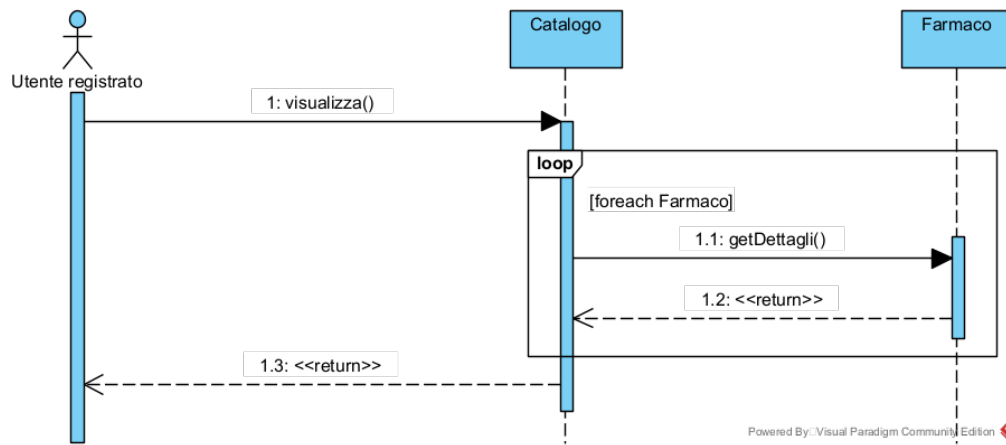


Figura 3.3: Diagramma di sequenza di analisi di VisualizzaCatalogo

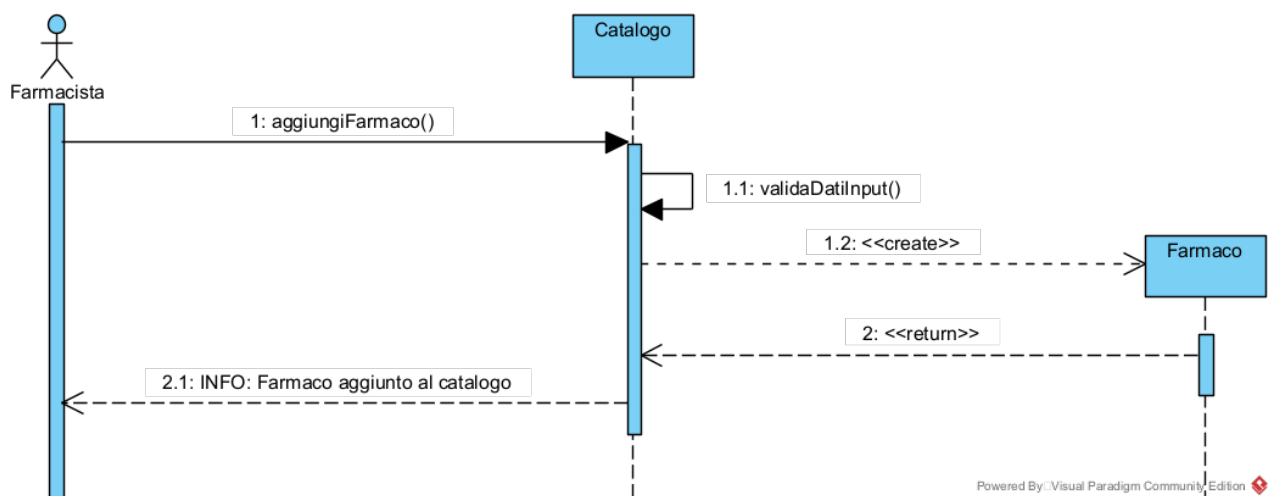


Figura 3.4: Diagramma di sequenza di analisi di AggiungiFarmaco

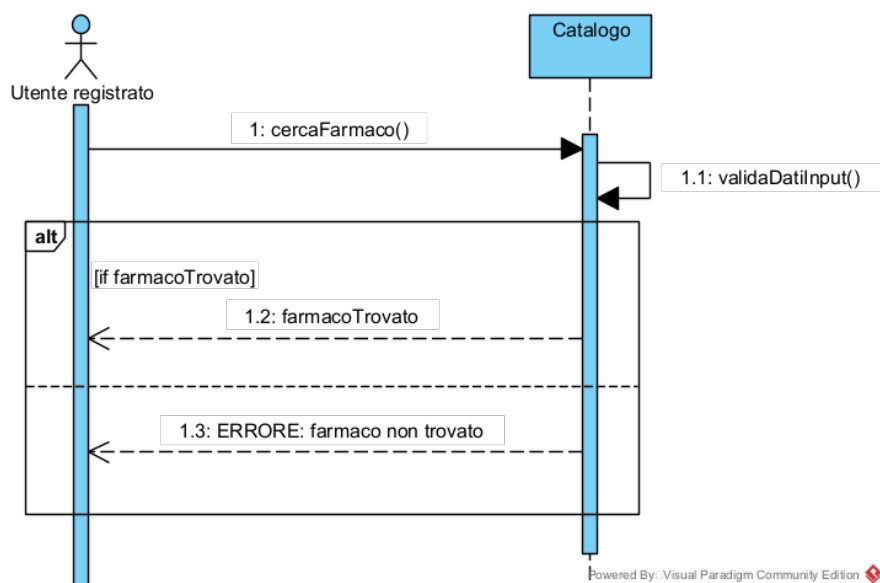


Figura 3.5: Diagramma di sequenza di analisi di CercaFarmaco

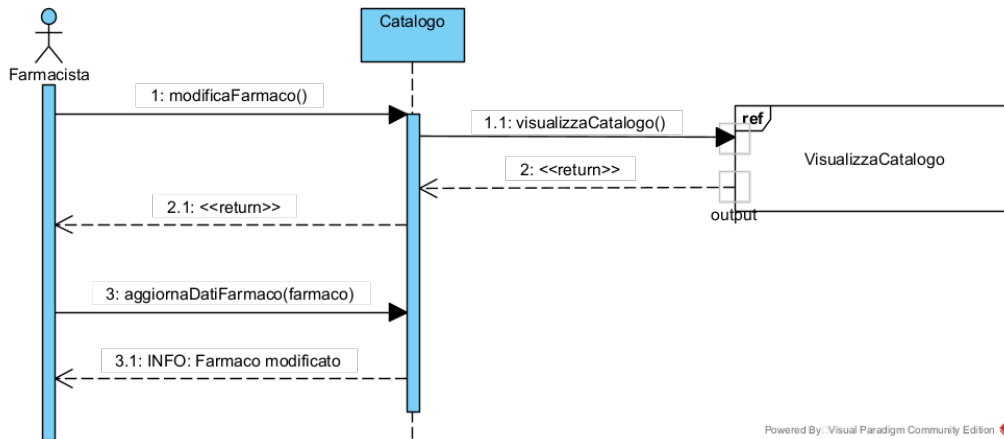


Figura 3.6: Diagramma di sequenza di analisi di ModificaFarmaco

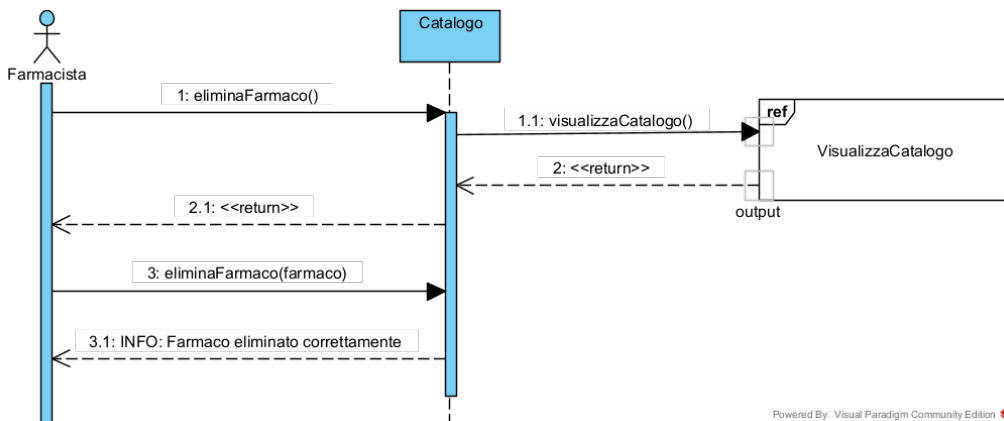


Figura 3.7: Diagramma di sequenza di analisi di EliminaFarmaco

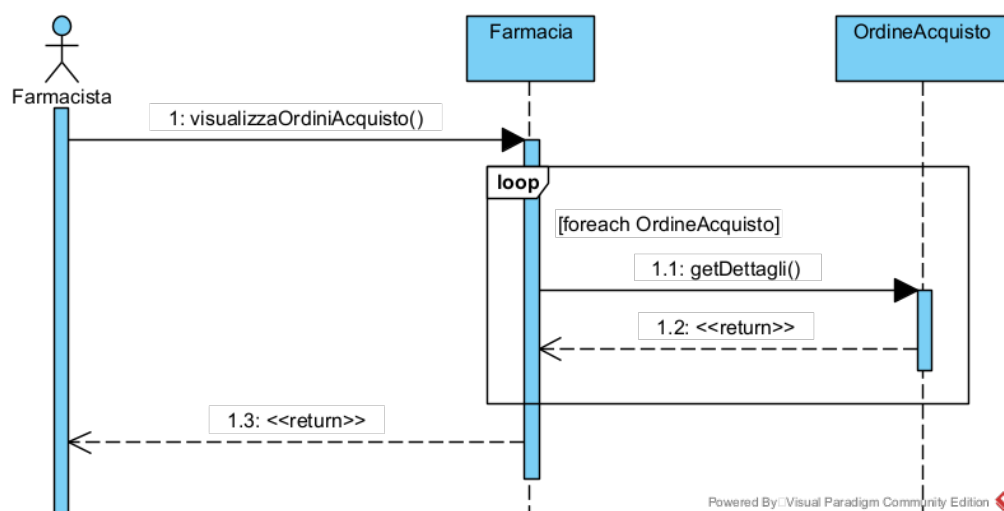


Figura 3.8: Diagramma di sequenza di analisi di VisualizzaOrdiniAcquisto

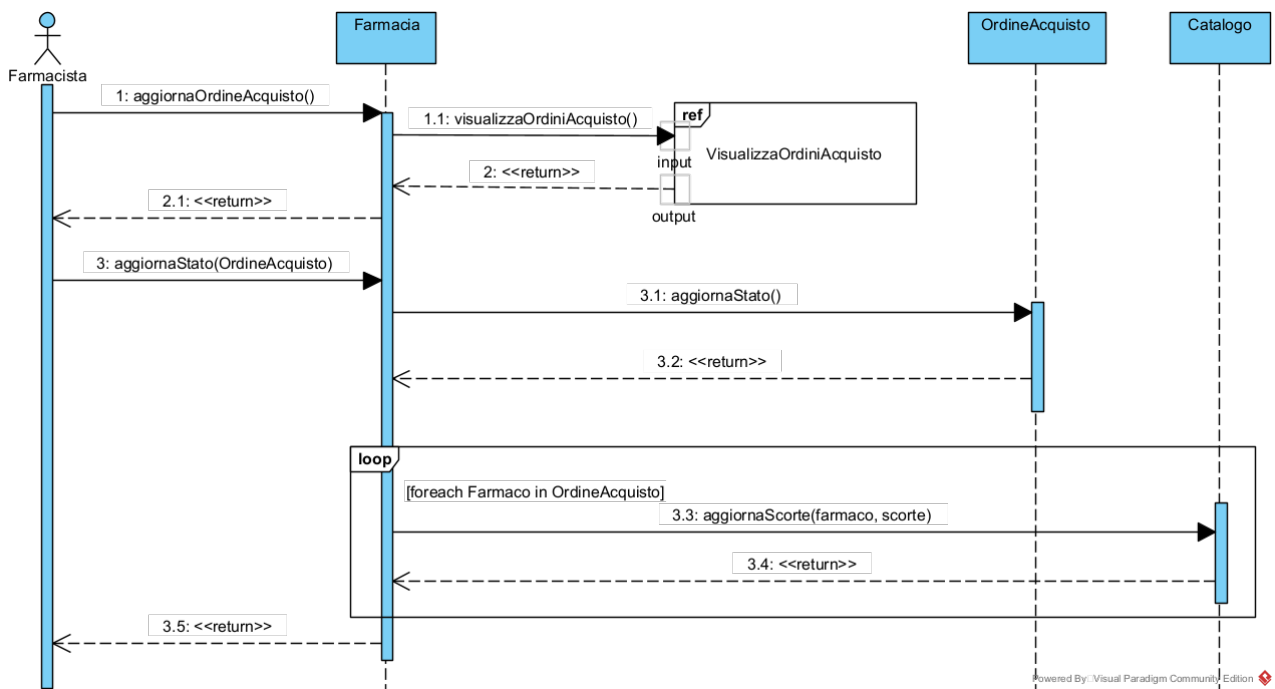


Figura 3.9: Diagramma di sequenza di analisi di AggiornaOrdineAcquisto (RegistraConsegnaOrdineAcquisto)

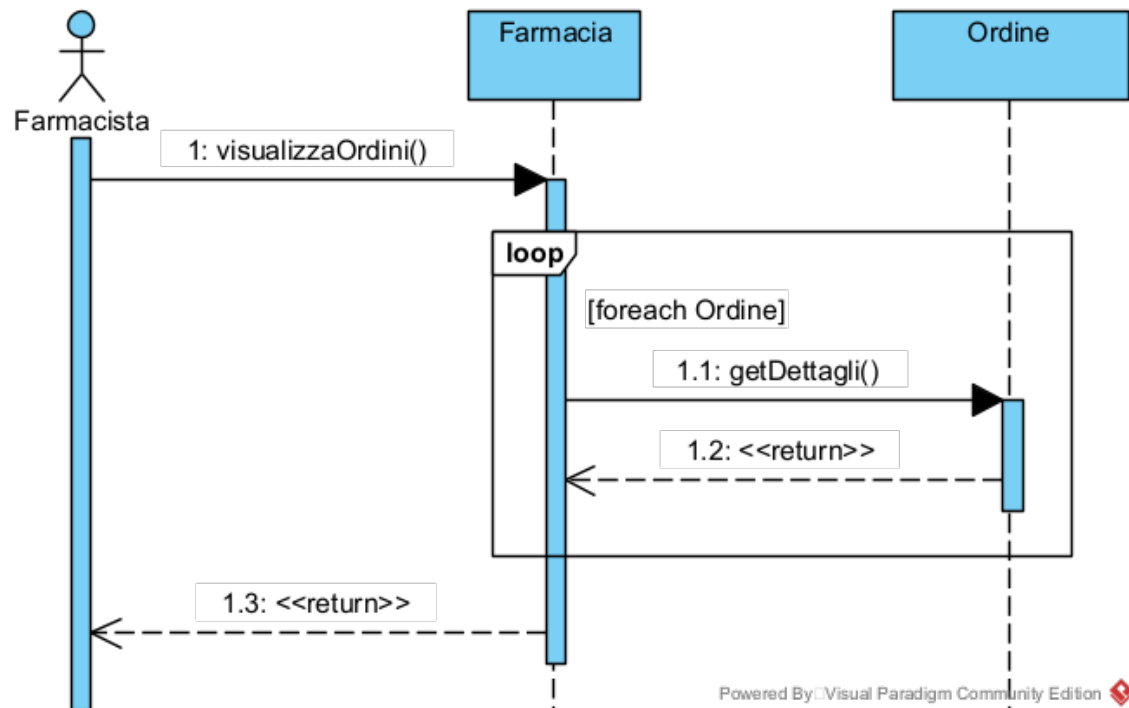


Figura 3.10: Diagramma di sequenza di analisi di VisualizzaOrdiniFarmacia

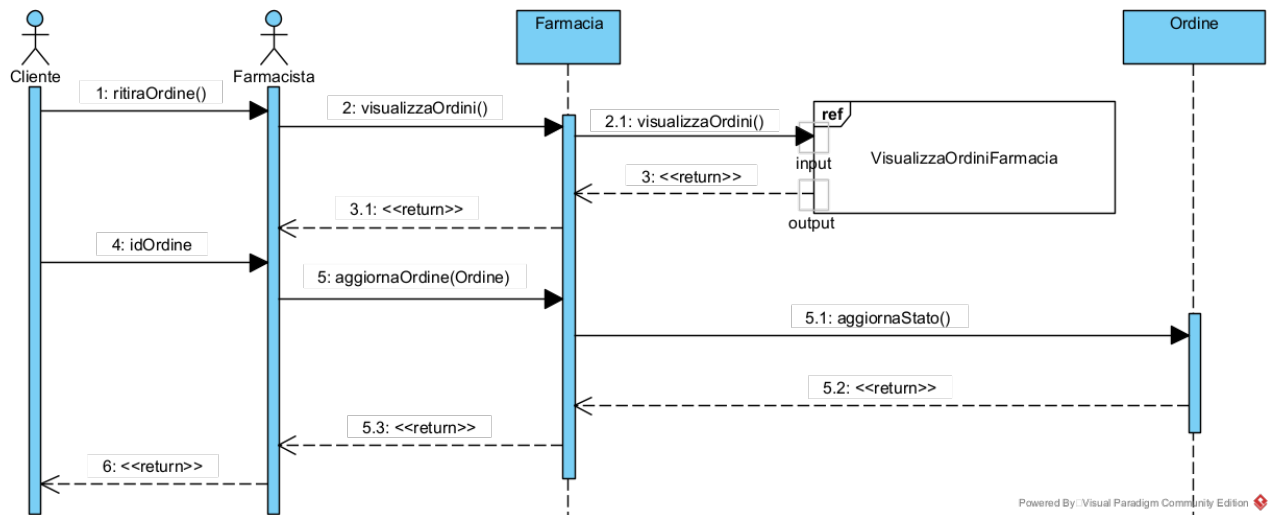


Figura 3.11: Diagramma di sequenza di analisi di RitiraOrdine

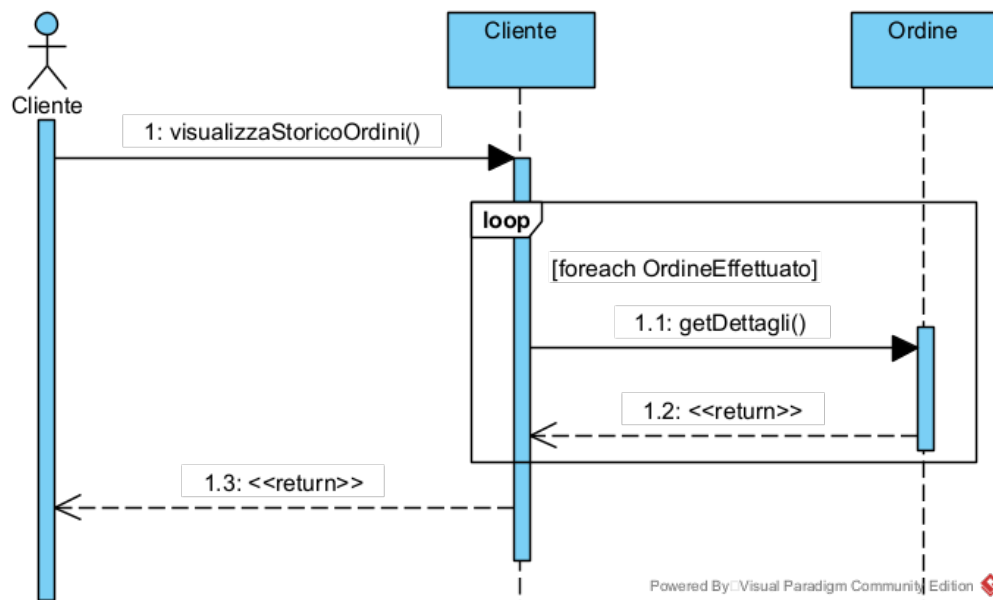


Figura 3.12: Diagramma di sequenza di analisi di VisualizzaStoricoOrdini

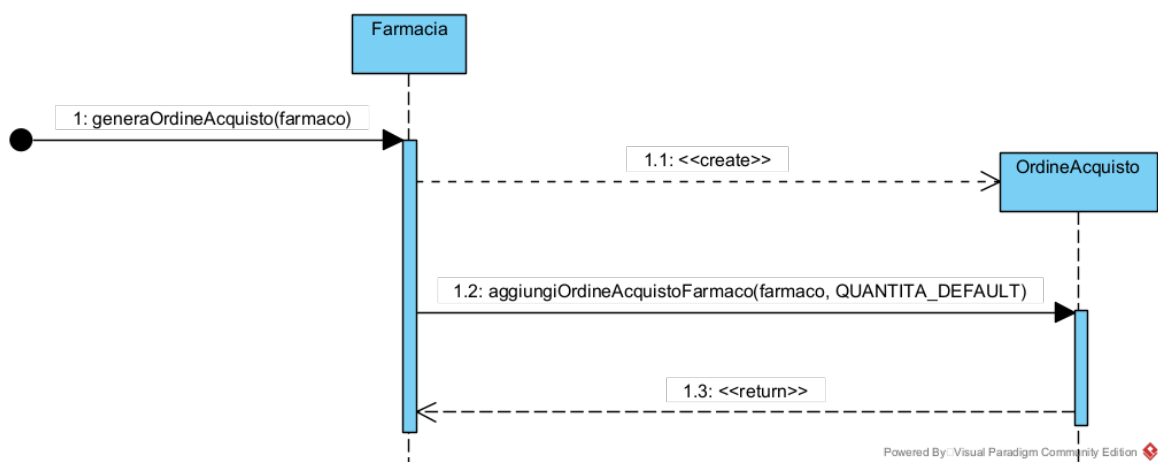


Figura 3.13: Diagramma di sequenza di analisi di GeneraOrdineAcquisto

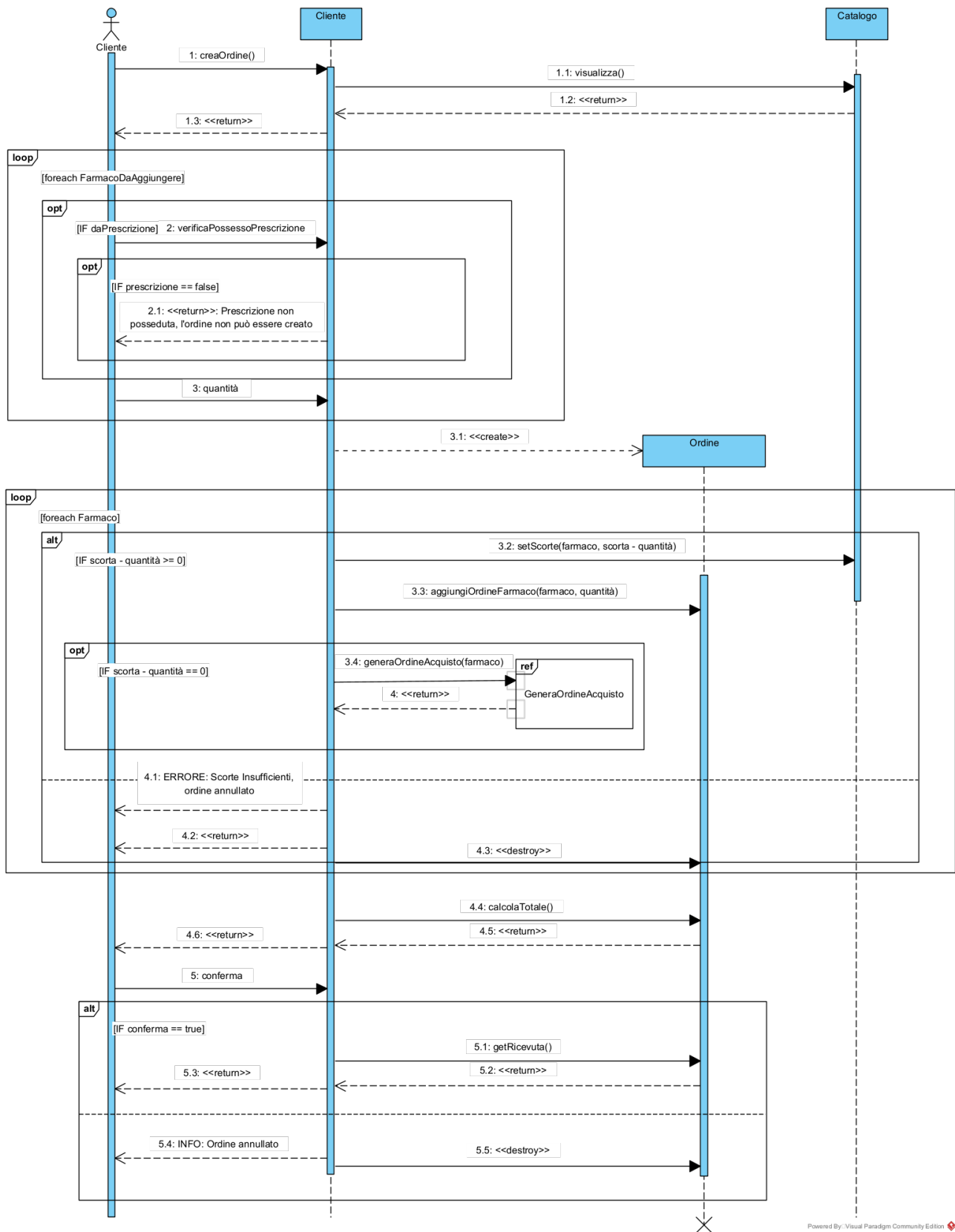


Figura 3.14: Diagramma di sequenza di analisi di CreaOrdine

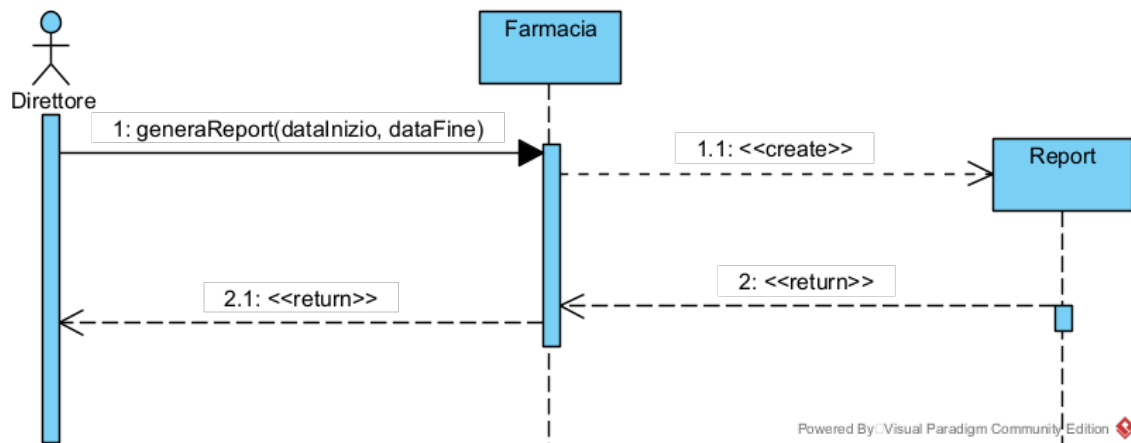


Figura 3.15: Diagramma di sequenza di analisi di GeneraReport

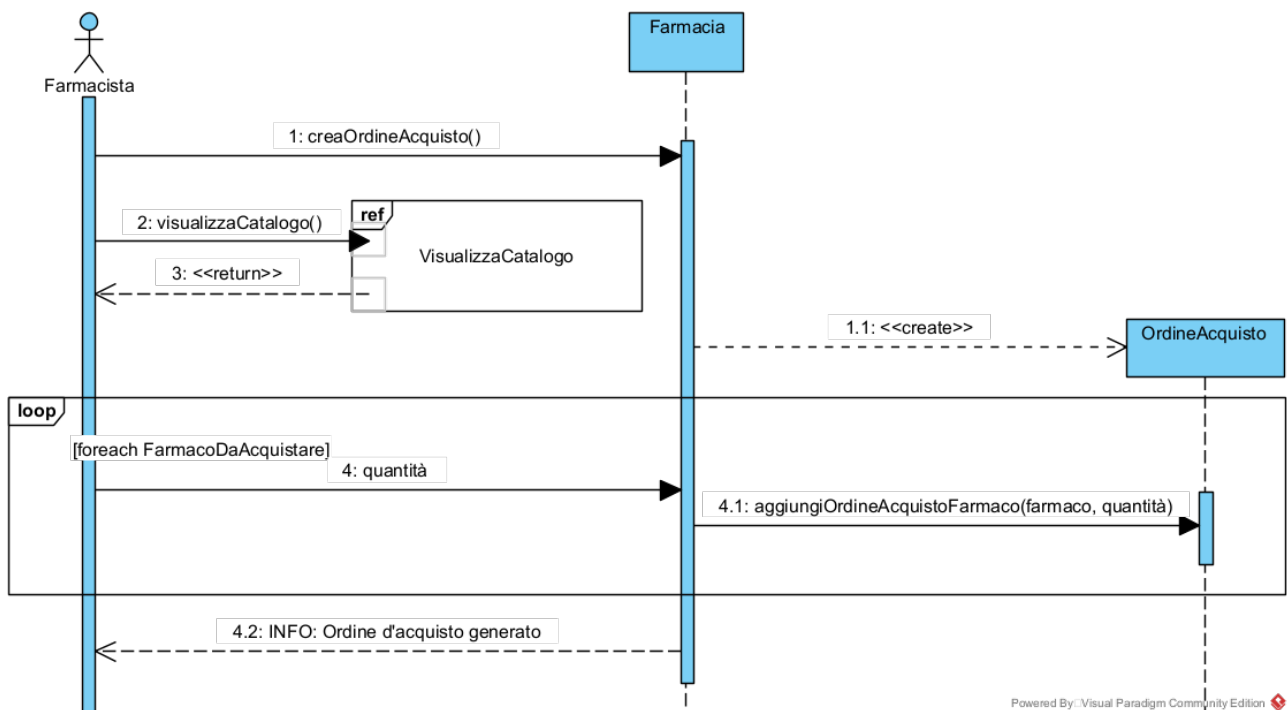


Figura 3.16: Diagramma di sequenza di analisi di GeneraOrdineAcquistoFarmacista

3.6 Verifica Completezza dei Requisiti

- RF₀₁, RF₀₆ Modellati in UCD da VisualizzaCatalogo con l'attore UtenteRegistrato
- RF₀₂ Modellato in UCD da AggiungiFarmaco con l'attore primario Farmacista
- RF₀₃ Modellato in UCD da ModificaFarmaco con l'attore primario Farmacista
- RF₀₄ Modellato in UCD da EliminaFarmaco con l'attore primario Farmacista
- RF₀₅ Modellato in UCD da RegistraCliente con l'attore primario Cliente non Registrato
- RF₀₇, RF₀₈, RF₀₉, RF₁₀, RF₁₇ Modellati in UCD da CreaOrdine con l'attore primario Cliente
- RF₁₁ Modellato in UCD da VisualizzaOrdiniAcquisto con l'attore primario Farmacista
- RF₁₂ Modellato in UCD da RegistraConsegnaOrdineAcquisto con l'attore primario Farmacista
- RF₁₃ Modellato in UCD da GeneraReport con l'attore primario Direttore
- RF₁₄ Modellato in UCD da VisualizzaOrdiniFarmacia con l'attore primario Farmacista
- RF₁₅ Modellato in UCD da RitiraOrdine con l'attore primario Cliente e attore secondario Farmacista
- RF₁₆ Modellato in UCD da VisualizzaStoricoOrdini con l'attore primario Cliente
- RF₁₈ Modellato in UCD da CercaFarmaco con l'attore primario UtenteRegistrato
- RF₁₉ Modellato in UCD da GeneraOrdineAcquistoFarmacista con l'attore primario Farmacista
- RD₀₁, RD₀₂ Modellato nel CD dalla classe Farmaco
- RD₀₃ Modellato nel CD dalla classe Cliente
- RD₀₄ Modellato nel CD dalla classe Report

UCD = Use Case Diagram

CD = Class Diagram

SD = Sequence Diagram

Capitolo 4

Piano di Test Funzionale

4.1 RegistraCliente

Category Partition Testing

Nome	Cognome	Username	Password	Email	DataNascita
<ul style="list-style-type: none">• Stringa di lunghezza ≤ 45• Stringa di lunghezza > 45 [ERROR]• Stringa vuota [ERROR]	<ul style="list-style-type: none">• Stringa di lunghezza ≤ 45• Stringa di lunghezza > 45 [ERROR]• Stringa vuota [ERROR]	<ul style="list-style-type: none">• Stringa di lunghezza ≤ 45• Stringa di lunghezza > 45 [ERROR]• Stringa vuota [ERROR]• Stringa già presente nel sistema [ERROR]	<ul style="list-style-type: none">• Stringa di lunghezza compresa tra 8 e 45• Stringa di lunghezza < 8 [ERROR]• Stringa di lunghezza > 45 [ERROR]	<ul style="list-style-type: none">• Stringa di lunghezza ≤ 45• Stringa di lunghezza > 45 [ERROR]• Stringa vuota [ERROR]• Stringa già presente nel sistema [ERROR]	<ul style="list-style-type: none">• Data con formato valido (gg-mm-aaaa)• Data con formato non valido [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $3 \cdot 3 \cdot 4 \cdot 3 \cdot 4 \cdot 2 = 864$.

Introduciamo i vincoli [ERROR]. Il numero di test da eseguire per testare singolarmente i vincoli è 13 (2 per Nome, 2 per Cognome, 3 per Username, 2 per Password, 3 per Email, 1 per DataNascita).

Il numero di test risultante è 14: $(1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1) + 13 = 14$.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input validi	Nome, Cognome, Username, Password, Email, DataNascita validi	Il cliente non è ancora registrato nel sistema	Nome: Mario Cognome: Rossi Username: matorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Registrazione effettuata	Il cliente è stato correttamente registrato nel sistema
2	Nome > 45 caratteri	Nome > 45 caratteri [ERROR], Cognome, Username, Password, Email, DataNascita validi	–	Nome: ... Cognome: Rossi Username: matorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Nome troppo lungo	–
3	Nome non specificato	Nome non specificato [ERROR], Cognome, Username, Password, Email, DataNascita validi	–	Nome: Cognome: Rossi Username: matorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Inserire un nome	–

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
4	Cognome > 45 caratteri	Cognome > 45 caratteri [ERROR], Nome, Username, Password, Email, DataNascita validi	–	Nome: Mario Cognome: ... Username: mariorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Cognome troppo lungo	–
5	Cognome non specificato	Cognome non specificato [ERROR], Nome, Username, Password, Email, DataNascita validi	–	Nome: Mario Cognome: Username: mariorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Inserire un cognome	–
6	Username > 45 caratteri	Username > 45 caratteri [ERROR], Nome, Cognome, Password, Email, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: ... Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Username troppo lungo	–
7	Username non specificato	Username non specificato [ERROR], Nome, Cognome, Password, Email, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Inserire uno username	–
8	Password < 8 caratteri	Password < 8 caratteri [ERROR], Nome, Cognome, Username, Email, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: mariorossi Password: prova Email: mario@gmail.com DataNascita: 22-06-1989	Password troppo corta	–
9	Password > 45 caratteri	Password > 45 caratteri [ERROR], Nome, Cognome, Username, Email, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: mariorossi Password: ... Email: mario@gmail.com DataNascita: 22-06-1989	Password troppo lunga	–
10	Email > 45 caratteri	Email > 45 caratteri [ERROR], Nome, Cognome, Username, Password, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: mariorossi Password: miapassword Email: ... DataNascita: 22-06-1989	Email troppo lunga	–
11	Email non specificato	Email non specificato [ERROR], Nome, Cognome, Username, Password, DataNascita validi	–	Nome: Mario Cognome: Rossi Username: Password: miapassword Email: DataNascita: 22-06-1989	Inserire un'email	–
12	Formato DataNascita non valido	Formato DataNascita non valido [ERROR], Nome, Cognome, Username, Password, Email validi	–	Nome: Mario Cognome: Rossi Username: mariorossi Password: miapassword Email: mario@gmail.com DataNascita: 1989-06-22	Formato data non valido	–

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
13	Username già presente nel sistema	Username già esistente [ERROR], Nome, Cognome, Password, DataNascita ed Email validi	Username “pippo2002” già presente nel sistema	Nome: Pippo Cognome: Baudo Username: pippo2002 Password: miapassword Email: pippo@gmail.com DataNascita: 1989-06-22	Username già utilizzato	–
14	Email già presente nel sistema	Email già esistente [ERROR], Nome, Cognome, Password, DataNascita e Username validi	Email “pippo@gmail.com” già presente nel sistema	Nome: Pippo Cognome: Baudo Username: pippo2002 Password: miapassword Email: pippo@gmail.com DataNascita: 1989-06-22	Email già utilizzata	–

4.2 LoginUtente

Category Partition Testing

Username	Password
<ul style="list-style-type: none"> Stringa di lunghezza ≤ 45 Stringa di lunghezza > 45 [ERROR] Stringa vuota [ERROR] Stringa valida ma non registrata nel sistema [ERROR] 	<ul style="list-style-type: none"> Stringa di lunghezza compresa tra 8 e 45 e corrispondente alla password dell'utente Stringa di lunghezza < 8 [ERROR] Stringa di lunghezza > 45 [ERROR] Stringa non corrispondente alla password dell'utente [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $4 \cdot 4 = 16$.

Introduciamo i vincoli [ERROR]. Il numero di test da eseguire per testare singolarmente i vincoli è 6 (3 per Username, 3 per Password).

Il numero di test risultante è 7: $(1 \cdot 1) + 6 = 7$.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input validi	Username, Password validi	L'utente deve essere correttamente registrato nel sistema	Username: mario Password: mia-password	Login effettuato	L'utente è entrato correttamente nel sistema
2	Username > 45 caratteri	Username > 45 caratteri [ERROR], Password valida	–	Username: ... Password: mia-password	Username troppo lungo	–
3	Username non specificato	Username non specificato [ERROR], Password valida	–	Username: Password: mia-password	Inserire uno username	–
4	Password < 8 caratteri	Password < 8 caratteri [ERROR], Username valido	–	Username: mario Password: prova	Password troppo corta	–
5	Password > 45 caratteri	Password > 45 caratteri [ERROR], Username valido	–	Username: mario Password: ...	Password troppo lunga	–
6	Password errata	Password errata [ERROR], Username valido	L'utente esiste nel sistema e ha come password 'passwd'	Username: mario Password: ciao	Password errata	–

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
7	Username non registrato	Username non registrato [ERROR]	L'utente non esiste nel sistema	Username: geronimo Password: stilton	Utente non registrato	–

4.3 AggiungiFarmaco

Category Partition Testing

Nome	Prezzo	Prescrizione	Scorte
<ul style="list-style-type: none"> Stringa di lunghezza ≤ 45 Stringa di lunghezza > 45 [ERROR] Stringa vuota [ERROR] Stringa già memorizzata [ERROR] 	<ul style="list-style-type: none"> Numero reale > 0 (€) Numero reale ≤ 0 (€) [ERROR] 	<ul style="list-style-type: none"> True (necessaria) / False (non necessaria) 	<ul style="list-style-type: none"> Numero intero ≥ 0 Numero intero < 0 [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $4 \cdot 2 \cdot 1 \cdot 2 = 16$.

Introduciamo i vincoli [ERROR]. Il numero di test da eseguire per testare singolarmente i vincoli è 5 (3 per Nome, 1 per Prezzo, 1 per Scorte).

Il numero di test risultante è 6: $(1 \cdot 1 \cdot 1 \cdot 1) + 5 = 6$.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input validi	Nome, Prezzo, Scorte e Prescrizione (sia True che False) validi	Il farmaco non è presente nel sistema	Nome: Rocefin Prezzo: 15.00 € Scorte: 60 Prescrizione: boolean	Farmaco aggiunto	Il farmaco viene correttamente aggiunto al catalogo
2	Nome > 45 caratteri	Nome > 45 caratteri [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	–	Nome: ... Prezzo: 16.10 € Scorte: 23 Prescrizione: boolean	Nome troppo lungo	–
3	Nome non specificato	Nome non specificato [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	–	Nome: Prezzo: 9.99 € Scorte: 10 Prescrizione: boolean	Inserire un nome	–
4	Prezzo ≤ 0 (€)	Prezzo ≤ 0 (€) [ERROR], Nome, Scorte e Prescrizione (sia True che False) validi	–	Nome: Brufen Prezzo: -8.00 € Scorte: 200 Prescrizione: boolean	Inserire un prezzo > 0 €	–
5	Scorte < 0	Scorte < 0 , Nome, Prezzo e Prescrizione (sia True che False) validi	–	Nome: Macladin Prezzo: 11.50 € Scorte: -36 Prescrizione: boolean	Inserire scorte ≥ 0	–
6	Nome già memorizzato	Nome già memorizzato [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	Esiste già un farmaco con il nome inserito	Nome: Cistalgan Prezzo: 19.90 € Scorte: 50 Prescrizione: boolean	Farmaco già esistente	–

4.4 ModificaFarmaco

Category Partition Testing

Nome	Prezzo	Prescrizione	Scorte
<ul style="list-style-type: none"> Stringa di lunghezza ≤ 45 Stringa di lunghezza > 45 [ERROR] Stringa vuota [ERROR] Stringa non presente nel sistema [ERROR] 	<ul style="list-style-type: none"> Numero reale > 0 (€) Numero reale ≤ 0 (€) [ERROR] 	<ul style="list-style-type: none"> True (necessaria) / False (non necessaria) 	<ul style="list-style-type: none"> Numero intero ≥ 0 Numero intero < 0 [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $4 \cdot 2 \cdot 2 \cdot 2 = 32$.

Introduciamo i vincoli [ERROR]. Il numero di test da eseguire per testare singolarmente i vincoli è 5 (3 per Nome, 1 per Prezzo, 1 per Scorte).

Il numero di test risultante è 7: $(1 \cdot 1 \cdot 2 \cdot 1) + 5 = 7$.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Tutti gli input validi, prescrizione True	Nome, Prezzo, Scorte e Prescrizione (True) validi	Il farmaco è presente nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: True	Modifica effettuata	Il farmaco viene correttamente modificato
2	Tutti gli input validi, prescrizione False	Nome, Prezzo, Scorte e Prescrizione (False) validi	Il farmaco è presente nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: False	Modifica effettuata	Il farmaco viene correttamente modificato
3	Nome > 45 caratteri	Nome > 45 caratteri [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	–	Nome: ... Prezzo: 22.00 € Scorte: 50 Prescrizione: boolean	Nome troppo lungo	–
4	Nome non specificato	Nome non specificato [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	–	Nome: Prezzo: 9.99 € Scorte: 44 Prescrizione: boolean	Inserire un nome	–
5	Prezzo ≤ 0 (€)	Prezzo ≤ 0 (€) [ERROR], Nome, Scorte e Prescrizione (sia True che False) validi	–	Nome: Augmentin Prezzo: -4.00 € Scorte: 10 Prescrizione: boolean	Inserire un prezzo > 0 €	–
6	Scorte < 0	Scorte < 0 , Nome, Prezzo e Prescrizione (sia True che False) validi	–	Nome: Neoduplamox Prezzo: 7.50 € Scorte: -8 Prescrizione: boolean	Inserire scorte ≥ 0	–
7	Il nome del farmaco non è presente nel sistema	Nome non presente nel sistema [ERROR], Prezzo, Scorte e Prescrizione validi	Il farmaco “Tachipirina” non esiste nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: False	Il farmaco non esiste	–

4.5 CercaFarmaco

Category Partition Testing

Nome
<ul style="list-style-type: none"> • Stringa di lunghezza ≤ 45 • Stringa di lunghezza > 45 [ERROR] • Stringa vuota [ERROR] • Nome di un farmaco non presente nel sistema [ERROR]

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 4.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Nome del farmaco valido	Nome del farmaco valido	Il farmaco è presente nel catalogo	Nome: Tachipirina	Il farmaco viene mostrato a video	–
2	Nome del farmaco > 45 caratteri	Nome del farmaco > 45 caratteri [ERROR]	–	Nome: ...	Nome del farmaco troppo lungo	–
3	Nome del farmaco non specificato	Nome del farmaco non specificato [ERROR]	–	Nome:	Inserire il nome di un farmaco	–
4	Il nome del farmaco non è presente nel sistema	Nome non presente nel sistema [ERROR]	Il farmaco con nome “Tachipirina” non esiste	Nome: Tachipirina	Il farmaco non esiste	–

4.6 EliminaFarmaco

Category Partition Testing

Nome
<ul style="list-style-type: none"> • Stringa di lunghezza ≤ 45 • Stringa di lunghezza > 45 [ERROR] • Stringa vuota [ERROR] • Stringa non presente nel sistema [ERROR]

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 4.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Nome del farmaco valido	Nome del farmaco valido	Il farmaco è presente nel catalogo	Nome: Tachipirina	Farmaco cancellato	Il farmaco viene cancellato dal catalogo
2	Nome del farmaco > 45 caratteri	Nome del farmaco > 45 caratteri [ERROR]	–	Nome: ...	Nome del farmaco troppo lungo	–
3	Nome del farmaco non specificato	Nome del farmaco non specificato [ERROR]	–	Nome:	Inserire il nome di un farmaco	–
4	Il farmaco non è presente nel sistema	Farmaco non presente nel sistema [ERROR]	Non esiste il farmaco chiamato “Tachipirina”	Nome: Tachipirina	Non puoi eliminare un farmaco che non esiste	–

4.7 GeneraReport

Category Partition Testing

DataInizio	DataFine
<ul style="list-style-type: none"> Data con formato valido (gg-mm-aaaa) Data con formato non valido [ERROR] 	<ul style="list-style-type: none"> Data con formato valido (gg-mm-aaaa) Data con formato non valido [ERROR] DataFine precede DataInizio [ERROR]

Il numero di test da effettuarsi senza particolari vincoli è: $2 \cdot 3 = 6$.

Introduciamo i vincoli [ERROR]. Il numero di test da eseguire per testare singolarmente i vincoli è 3 (1 per DataInizio, 2 per DataFine).

Il numero di test risultante è 4: $(1 \cdot 1) + 3 = 4$.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Input	Output Attesi
1	Tutti gli input validi	DataInizio, DataFine validi	DataInizio: 01-06-2024 DataFine: 31-06-2024	Il report viene generato
2	Formato DataInizio non valido	Formato DataInizio non valido [ERROR]	DataInizio: 2024-06-01 DataFine: 31-06-2024	Formato DataInizio non valido
3	Formato DataFine non valido	Formato DataFine non valido [ERROR]	DataInizio: 01-06-2024 DataFine: 2024-06-31	Formato DataFine non valido
4	DataFine precede DataInizio	DataFine precede DataInizio [ERROR]	DataInizio: 31-06-2024 DataFine: 01-06-2024	Intervallo di date errato

4.8 GeneraOrdineAcquistoFarmacista

Farmaci-Quantità
<ul style="list-style-type: none"> Lista di coppie (nomeFarmaco, quantità) non vuota. Per ogni nomeFarmaco esiste un corrispondente farmaco nel sistema. Lista vuota [ERROR]

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 2.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Ordine valido	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort'	[('Tachipirina', 5), ('Fluifort', 10)]	Ordine di acquisto generato	Un ordine di acquisto viene correttamente creato
2	Ordine vuoto	Lista vuota [ERROR]	—	—	Non puoi creare un ordine di acquisto vuoto	—

4.9 RegistraConsegnaOrdineAcquisto

Category Partition Testing

Id dell'ordine di acquisto
<ul style="list-style-type: none"> • Stringa di lunghezza ≤ 45 • Stringa di lunghezza > 45 [ERROR] • Stringa vuota [ERROR] • Stringa non presente nel sistema [ERROR]

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 4.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Id dell'ordine di acquisto valido	Id dell'ordine di acquisto valido	–	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	Ordine ricevuto	Lo stato dell'ordine di acquisto e le scorte in magazzino sono aggiornati
2	Id > 45 caratteri	Id > 45 caratteri [ERROR]	–	Id: ...	Id troppo lungo	–
3	Id non specificato	Id non specificato [ERROR]	–	Id:	Inserire l'Id di un ordine di acquisto	–
4	Id non presente nel sistema	Id non presente nel sistema [ERROR]	Non esiste un ordine di acquisto con Id "5ea930bc-f0a5-427a-8ca1-f9a2a6146948"	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	L'ordine di acquisto non esiste	–

4.10 RitiraOrdine

Category Partition Testing

idOrdine
<ul style="list-style-type: none"> • Stringa di lunghezza ≤ 45 • Stringa di lunghezza > 45 [ERROR] • Stringa vuota [ERROR] • Stringa non presente nel sistema [ERROR]

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 4.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Id dell'ordine valido	Id dell'ordine valido	–	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	Stato ordine cambiato con successo	Lo stato dell'ordine è stato cambiato con successo
2	Id dell'ordine > 45 caratteri	Id dell'ordine > 45 caratteri [ERROR]	–	Id: ...	Id dell'ordine troppo lungo	–

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
3	Id dell'ordine non specificato	Id dell'ordine non specificato [ERROR]	–	Id:	Inserire l'ID di un ordine	–
4	Id non presente nel sistema	Id non presente nel sistema [ERROR]	Non esiste un farmaco con id "5ea930bc-f0a5-427a-8ca1-f9a2a6146948"	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	L'ordine non esiste	–

4.11 CreaOrdine

Farmaci-Quantità
<ul style="list-style-type: none"> • Lista di coppie (nomeFarmaco, quantità) non vuota. Per ogni nomeFarmaco esiste un corrispondente farmaco nel sistema ed ogni quantità richiesta è < delle scorte in magazzino. • Lista di coppie (nomeFarmaco, quantità) non vuota. Per ogni nomeFarmaco esiste un corrispondente farmaco nel sistema ed esiste almeno un farmaco la cui quantità richiesta è esattamente uguale alle scorte in magazzino. • Lista vuota [ERROR] • Lista di coppie (nomeFarmaco, quantità). Per ogni nomeFarmaco esiste un corrispondente farmaco nel sistema ed esiste almeno un farmaco la cui quantità richiesta è > delle scorte in magazzino [ERROR].

Essendo previsto un solo input, il numero di test da effettuarsi è pari a 4.

Test Suite

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese
1	Ordine valido, l'ordine non esaurisce le scorte di nessun farmaco	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 5), ('Fluifort', 10)]	Ordine generato	Ordine creato: scorte in magazzino decrementate
2	Ordine valido, l'ordine esaurisce le scorte di un farmaco	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 80), ('Fluifort', 10)]	Ordine generato	Ordine creato. Le scorte in magazzino vengono correttamente decrementate e viene generato un ordine di fornitura per il farmaco esaurito, richiedendone una quantità di default
3	Ordine vuoto	Lista vuota [ERROR]	–	–	Non puoi creare un ordine vuoto	–
4	Ordine invalido per scorte insufficienti	Scorte insufficienti [ERROR]	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 5), ('Fluifort', 150)]	Ordine non creato per scorte insufficienti	–

Capitolo 5

Progettazione

5.1 Diagramma delle classi

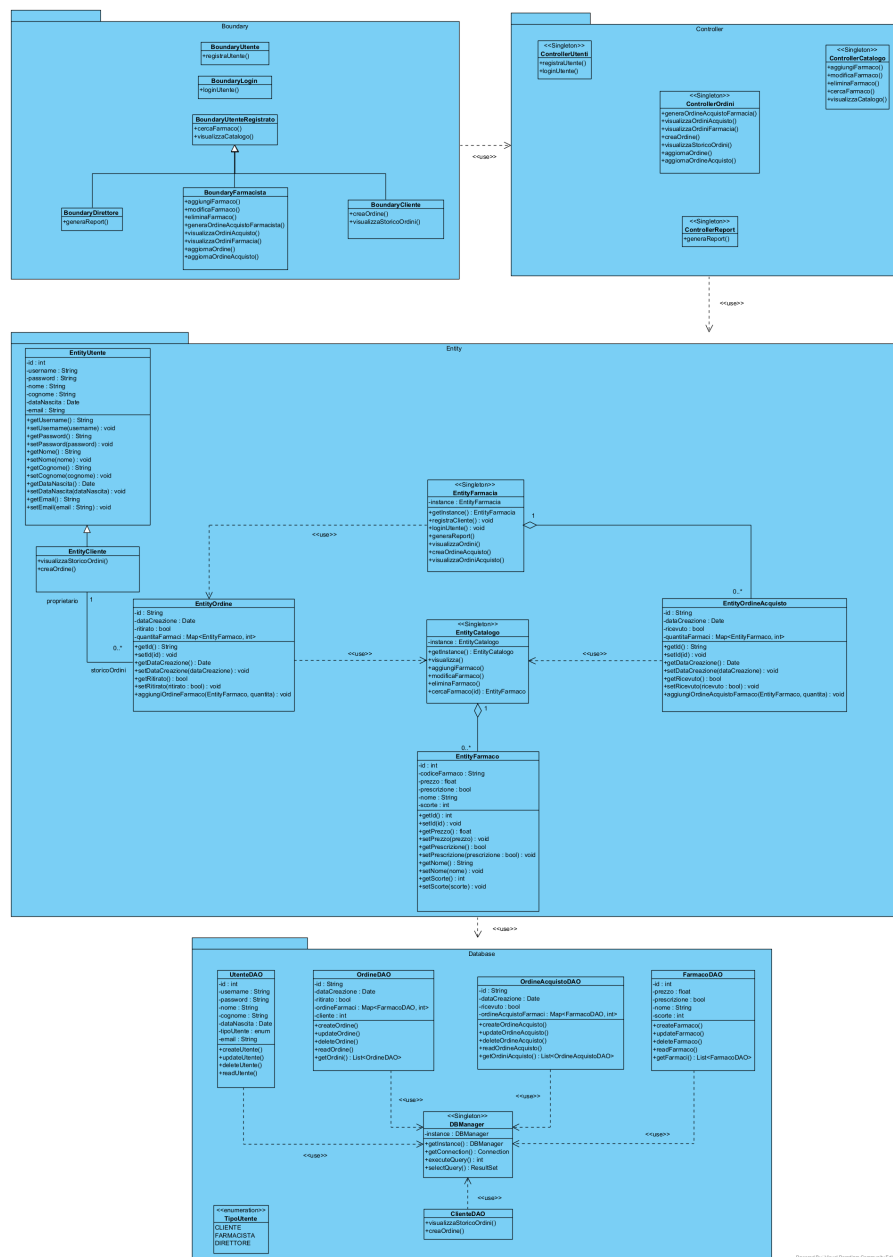


Figura 5.1: Diagramma delle classi di progettazione

5.2 Modello Entity-Relationship del Database

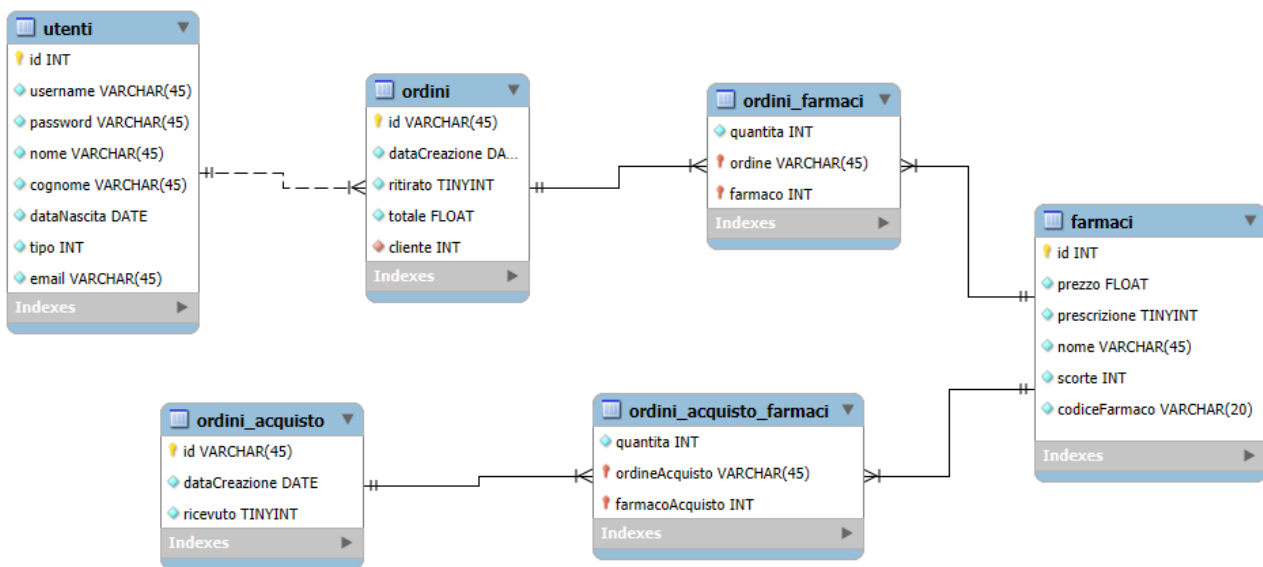


Figura 5.2: Modello E/R del Database

5.3 Diagrammi di sequenza

Capitolo 6

Implementazione

6.1 Specifiche di implementazione Java

Riportiamo di seguito le classi Java utilizzate, con la loro suddivisione in package:

Package `farmacia.boundary`

- `RegisterPage`
- `LoginPage`
- `HomePageCliente`
- `VisualizzaCatalogoPage`
- `FarmaciTableModel`
- `CreaOrdinePage`
- `VisualizzaStoricoOrdiniPage`
- `CercaFarmacoPage`
- `HomePageFarmacista`
- `AggiungiFarmacoPage`
- `ModificaFarmacoPage`
- `EliminaFarmacoPage`
- `VisualizzaCatalogoFarmacistaPage`
- `VisualizzaOrdiniClientiPage`
- `VisualizzaOrdiniAcquistoPage`
- `GeneraOrdineAcquistoPage`
- `ScegliFarmacoModificaPage`
- `RegistraRitiroOrdineClientePage`
- `RegistraConsegnaOrdineAcquistoPage`
- `HomePageDirettore`
- `GeneraReportPage`

Package `farmacia.boundary.creaordine`

- `CreaOrdinePage`
- `OrdineTableModel`
- `OrdineTableColumnModel`
- `QuantitaCellEditor`

Package `farmacia.boundary.datepicker`

- `DatePanel`
- `DateTextField`

Package `farmacia.dto`

- `DTO`

Package `farmacia.controller`

- `ControllerCatalogo`
- `ControllerOrdini`
- `ControllerUtenti`
- `ControllerReport`

Package `farmacia.database`

- `DBManager`
- `UtenteDAO`
- `FarmacoDAO`
- `OrdineDAO`
- `OrdineAcquistoDAO`

Package `farmacia.test.controllercatalogo`

- `AggiungiFarmacoTest`
- `CercaFarmacoTest`
- `CercaFarmacoInesistenteTest`
- `EliminaFarmacoTest`
- `ModificaFarmacoTest`
- `ModificaFarmacoInesistenteTest`

Package `farmacia.test.controllerordini`

- `CreaOrdineTest`
- `VisualizzaStoricoOrdiniTest`
- `AggiornaOrdineTest`
- `AggiornaOrdineAcquistoFarmaciaTest`
- `GeneraOrdineAcquistoFarmacistaTest`

Package `farmacia`

- `Main`
- `TipoUtente`

`farmacia.entity`

- `EntityCatalogo`
- `EntityCliente`
- `EntityUtente`
- `EntityFarmacia`
- `EntityFarmaco`
- `EntityOrdine`
- `EntityOrdineAcquisto`
- `Sessione`

Package `farmacia.exceptions`

- `DBException`
- `LoginFailedException`
- `RegistrationFailedException`
- `FarmacoCreationFailedException`
- `FarmacoNotFoundException`
- `OrderCreationFailedException`
- `OrderNotFoundException`
- `ReportException`

Package `farmacia.test.controllerutenti`

- `LoginUtenteTest`
- `RegistraUtenteTest`

Package `farmacia.testcontrollerreport`

- `GeneraReportTest`

Le linee di codice scritte sono:

- LOC = 6041
- LLOC = 4257

6.2 Artefatti necessari per l'installazione e l'esecuzione

Le dipendenze necessarie al fine di eseguire l'applicativo sono:

- OpenJDK-22.0.1
- Oracle MySQL Server 8.0.38 (con il file db_farmacia.sql)
- mysql-connector-j-8.4.0.jar
- junit-4.10.jar
- bluecatcode.junit.4.10.extended.jar (JUnit)

6.3 Documentazione

La documentazione del sistema è presente all'interno della cartella `javadoc/`.

6.4 Diagramma di deployment

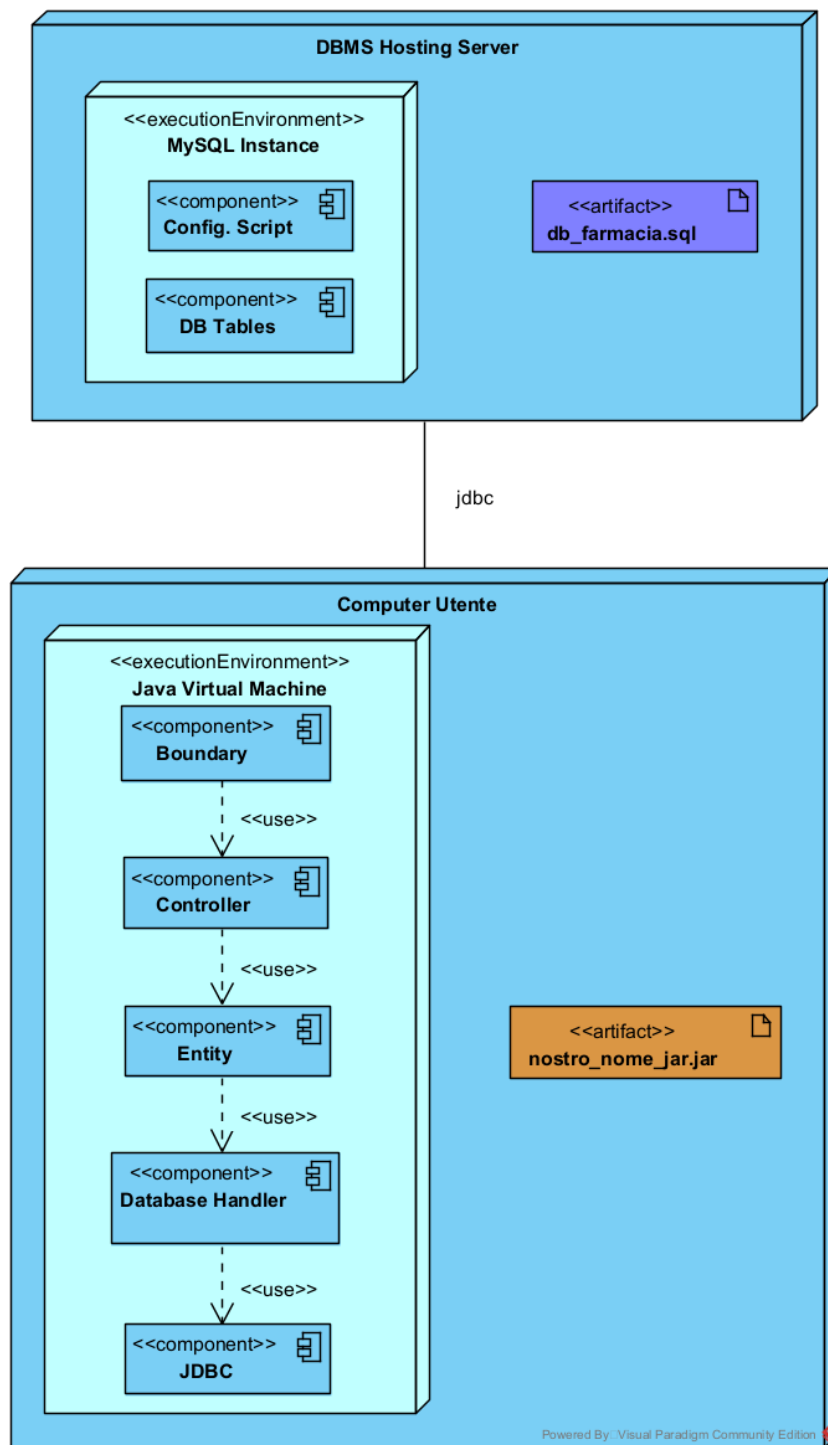


Figura 6.1: Diagramma di deployment

Capitolo 7

Testing

7.1 Test strutturale

7.1.1 creaOrdine()

Codice Java

```
1 public String creaOrdine(Map<Integer, Integer> farmaciQuantita) throws
   ↳ OrderCreationFailedException {
2     if (farmaciQuantita.isEmpty())
3         throw new OrderCreationFailedException("Ordine vuoto");
4
5     EntityCatalogo catalogo = EntityCatalogo.getInstance();
6     if (!catalogo.checkScorte(farmaciQuantita)) {
7         throw new OrderCreationFailedException("Ordine non creato per mancanza scorte");
8     }
9     EntityOrdine ordine = new EntityOrdine(this.getId());
10    try {
11        EntityOrdineAcquisto ordineAcquisto = new EntityOrdineAcquisto();
12        for (Map.Entry<Integer, Integer> entry : farmaciQuantita.entrySet()) {
13            int id = entry.getKey();
14            int quantita = entry.getValue();
15            EntityFarmaco farmaco = catalogo.cercaFarmacoById(id);
16            ordine.aggiungiOrdineFarmaco(farmaco, quantita);
17            int scorteResidue = catalogo.decrementaScorte(id, quantita);
18            if (scorteResidue == 0) {
19                ordineAcquisto.aggiungiOrdineAcquistoFarmaco(farmaco,
20                    ↳ EntityOrdineAcquisto.QUANTITA_ORDINE_DEFAULT);
21            }
22        }
23        if (!ordineAcquisto.getQuantitaFarmaci().isEmpty()) {
24            ordineAcquisto.salvaInDB();
25        }
26        ordine.salvaInDB();
27        storicoOrdini.add(ordine);
28        return ordine.getId();
29    } catch (FarmacoNotFoundException e) {
30        throw new OrderCreationFailedException("Errore creazione ordine, farmaco non
31            ↳ trovato");
32    } catch (DBException e) {
33        throw new OrderCreationFailedException(e.getMessage());
34    }
```

Control Flow Graph

Il numero di cammini linearmente indipendenti è detto *numero cicломatico* di McCabe, e può essere calcolato equivalentemente in uno dei modi seguenti. Sia G il grafo della funzione, allora risulta:

1. $V(G) = E - N + 2$ in cui $E = \# \text{archi in } G$, $N = \# \text{nodi in } G$
2. $V(G) = P + 1$ con $P = \# \text{predicati in } G$
3. $V(G) = R + 1$ con $R = \# \text{regioni chiuse in } G$

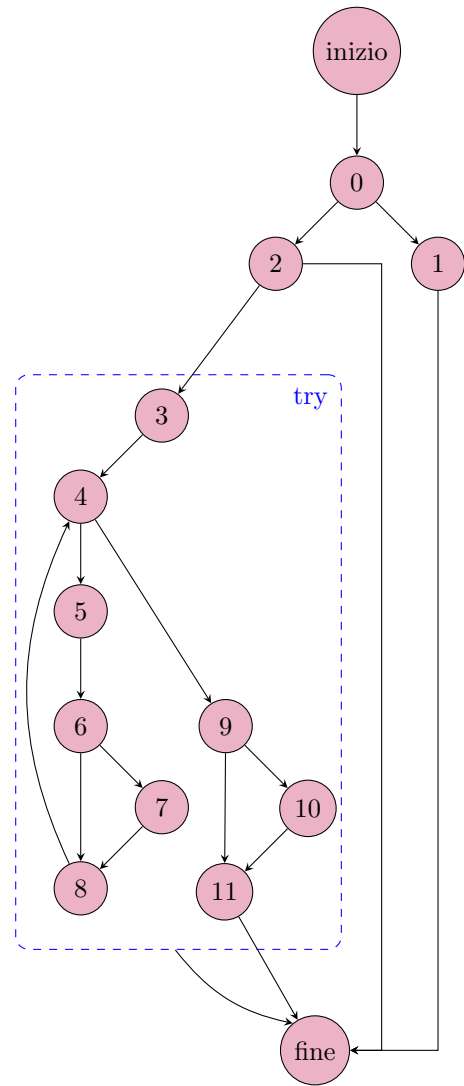
Nel nostro caso:

- $E = 16$
- $N = 12$
- $P = 5$
- $R = 5$

1. $V(G) = E - N + 2 = 16 - 12 + 2 = 6$
2. $P + 1 = 5 + 1 = 6$
3. $R + 1 = 5 + 1 = 6$

I cammini di base sono:

1. 0-1
2. 0-2
3. 0-2-3-4-9-11
4. 0-2-3-4-9-10-11
5. 0-2-3-4-5-6-8-4-9-11
6. 0-2-3-4-5-6-7-8-4-9-10-11



Test suite strutturale

Test Case ID	Descrizione	Cammino Coperto	Pre-condizioni	Input	Esito
1	Ordine vuoto	0-1	–	[]	Viene lanciata l'eccezione <code>OrderCreationFailedException("Ordine vuoto")</code>
2	Scorte insufficienti	0-2	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 15)]	Viene lanciata l'eccezione <code>OrderCreationFailedException("Ordine non creato per mancanza scorte")</code>
3	Ordine contemporaneamente vuoto e non vuoto	0-2-3-4-9-11	–	[]	Cammino non percorribile
4	Ordine vuoto, viene generato un ordine d'acquisto	0-2-3-4-9-10-11	–	[]	Cammino non percorribile
5	Ordine creato, senza generare un ordine d'acquisto	0-2-3-4-5-6-8-4-9-11	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 5)]	Viene creato un ordine
6	Ordine creato, si genera un ordine d'acquisto	0-2-3-4-5-6-7-8-4-9-10-11	Sono presenti 10 scorte del farmaco 'Tachipirina'	[(⟨idTachipirina⟩, 10)]	Ordine creato, parte una richiesta di fornitura per 50 'Tachipirina'

L'id dei farmaci viene scelto dal DB all'atto dell'aggiunta (indice autoincrementale). Pertanto, nel test l'id del farmaco viene ricavato con una funzione esterna.

7.1.2 modificaFarmaco()

Codice Java

```

1  public void modificaFarmaco(int id, float prezzo, boolean prescrizione, String nome, int
   ↪  scorte) throws FarmacoNotFoundException {
2      try {
3          FarmacoDAO.aggiornaFarmacoDB(id, prezzo, prescrizione, nome, scorte);
4          for (EntityFarmaco farmaco : farmaci) {
5              if (farmaco.getId() == id) {
6                  farmaco.setPrezzo(prezzo);
7                  farmaco.setPrescrizione(prescrizione);
8                  farmaco.setNome(nome);
9                  farmaco.setScorte(scorte);
10             }
11         }
12     } catch (DBException e) {
13         throw new FarmacoNotFoundException(e.getMessage());
14     }
15 }

```

Control Flow Graph

Il numero di cammini linearmente indipendenti è detto *numero cicломatico* di McCabe, e può essere calcolato equivalentemente in uno dei modi seguenti. Sia G il grafo della funzione, allora risulta:

1. $V(G) = E - N + 2$ in cui $E = \#$ archi in G , $N = \#$ nodi in G
2. $V(G) = P + 1$ con $P = \#$ predicati in G
3. $V(G) = R + 1$ con $R = \#$ regioni chiuse in G

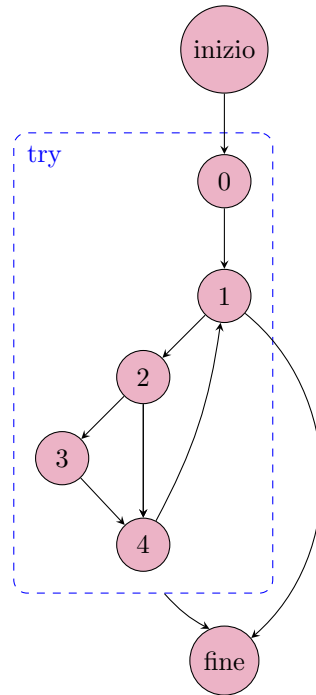
Nel nostro caso:

- $E = 6$
- $N = 5$
- $P = 2$
- $R = 2$

1. $V(G) = E - N + 2 = 6 - 5 + 2 = 3$
2. $P + 1 = 2 + 1 = 3$
3. $R + 1 = 2 + 1 = 3$

I cammini di base sono:

1. 0-1
2. 0-1-2-4-1
3. 0-1-2-3-4-1



Test suite strutturale

Test Case ID	Descrizione	Cammino Coperto	Pre-condizioni	Input	Esito
1	Il farmaco è presente nel DB ma non nella collection locale, che risulta vuota	0-1	Il farmaco è presente nel DB ma non nella collection locale	–	Cammino non percorribile: la collection locale è sempre sincronizzata con il DB
2	La collection locale ha un solo farmaco che non è quello da modificare	0-1-2-4-1	–	–	Cammino non percorribile: se il farmaco è presente nel DB, deve essere presente anche nella collection locale
3	Il farmaco esiste	0-1-2-3-4-1	Il farmaco esiste	Nome: Plasil Prezzo: 11.50 € Scorte: 30 Prescrizione: True	Farmaco modificato

7.2 Test funzionale

In questa sezione ci occupiamo di riportare i risultati associati alla test suite definita nel Capitolo 4. Di seguito sono riportati gli esiti dell'esecuzione dei vari test case sull'implementazione Java del servizio richiesto: in particolare non vengono mostrati i casi di test basati esclusivamente sull'input validation, effettuata completamente a livello **Boundary**. I test riportati, dunque, sono eseguiti a livello **Controller**.

I test sono stati implementati utilizzando il framework **JUnit**. Per la corretta esecuzione dei test è necessario eseguire nel pannello di controllo dell'istanza **MySQL** lo script `svuota_db_pre_test.sql`, che svuota tutte le tabelle del database nell'ordine corretto (per rispettare i vincoli inter-relazionali), oltre a registrare un account da farmacista ed uno da direttore. È necessario, inoltre, disabilitare la **SAFE UPDATE/DELETE MODE** di **MySQL**.

7.2.1 RegistraCliente

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Tutti gli input validi	Nome, Cognome, Username, Password, Email, DataNascita validi	Il cliente non è ancora registrato nel sistema	Nome: Mario Cognome: Rossi Username: mariorossi Password: miapassword Email: mario@gmail.com DataNascita: 22-06-1989	Registrazione effettuata	Il cliente è stato correttamente registrato nel sistema	PASS
13	Username già presente nel sistema	Username già esistente [ERROR], Nome, Cognome, Password, DataNascita ed Email validi	Username "pippo2002" già presente nel sistema	Nome: Pippo Cognome: Baudo Username: pippo2002 Password: miapassword Email: pippo@gmail.com DataNascita: 1989-06-22	Username già utilizzato	–	PASS
14	Email già presente nel sistema	Email già esistente [ERROR], Nome, Cognome, Password, DataNascita e Username validi	Email "pippo@gmail.com" già presente nel sistema	Nome: Pippo Cognome: Baudo Username: pippo2002 Password: miapassword Email: pippo@gmail.com DataNascita: 1989-06-22	Email già utilizzata	–	PASS

7.2.2 LoginUtente

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Tutti gli input validi	Username, Password validi	L'utente deve essere correttamente registrato nel sistema	Username: mario Password: mia-password	Login effettuato	L'utente è entrato correttamente nel sistema	PASS
6	Password errata	Password errata [ERROR], Username valido	L'utente esiste nel sistema e ha come password 'passwd'	Username: mario Password: ciao	Password errata	–	PASS
7	Username non registrato	Username non registrato [ERROR]	L'utente non esiste nel sistema	Username: geronimo Password: stilton	Utente non registrato	–	PASS

7.2.3 AggiungiFarmaco

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Tutti gli input validi	Nome, Prezzo, Scorte e Prescrizione (sia True che False) validi	Il farmaco non è presente nel sistema	Nome: Rocefin Prezzo: 15.00 € Scorte: 60 Prescrizione: boolean	Farmaco aggiunto	Il farmaco viene correttamente aggiunto al catalogo	FAIL
6	Nome già memorizzato	Nome già memorizzato [ERROR], Prezzo, Scorte e Prescrizione (sia True che False) validi	Esiste già un farmaco con il nome inserito	Nome: Cistalgan Prezzo: 19.90 € Scorte: 50 Prescrizione : boolean	Farmaco già esistente	–	PASS

Il test case 1 per la funzionalità AggiungiFarmaco ha dato come esito **FAIL**, generando un'eccezione a livello database. Abbiamo riscontrato la causa del malfunzionamento nel seguente snippet di codice presente nella classe `FarmacoDAO`:

```

1 String query = String.format("INSERT INTO farmaci (prezzo, prescrizione, nome, scorte,
  ↳ codice) " + "VALUES (%.2f, %d, '%s', %d, '%s');",
2     this.prezzo, this.prescrizione ? 1 : 0, this.nome, this.scorte, this.codice
3     );

```

Il problema è stato individuato a livello del metodo `String.format(...)`, che stampa le variabili `float` utilizzando la virgola come separatore tra parte intera e parte decimale (per la convenzione geografica), dunque qualsiasi numero decimale passato nel parametro `prezzo` viene diviso, a livello `SQL`, in due parametri. Il corpo della chiamata è stato modificato come segue:

```

1 String query = String.format(Locale.US, "INSERT INTO farmaci (prezzo, prescrizione, nome,
  ↳ scorte, codice) " + "VALUES (%.2f, %d, '%s', %d, '%s');",
2     this.prezzo, this.prescrizione ? 1 : 0, this.nome, this.scorte, this.codice
3     );

```

Tale problematica è presente in ogni metodo delle classi `DAO` che esegue query `SQL` contenenti un parametro di tipo `float`. La documentazione di `java.util.String` riporta che è possibile forzare la rappresentazione geografica dei numeri floating point utilizzando la costante `Locale.US`: con questo accorgimento, tutti i test che presentano questa problematica restituiscono esito positivo.

7.2.4 CercaFarmaco

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Nome del farmaco valido	Nome del farmaco valido	Il farmaco è presente nel catalogo	Nome: Tachipirina	Il farmaco viene mostrato a video	–	PASS
4	Nome di un farmaco che non esiste	Nome di un farmaco che non esiste [ERROR]	Il farmaco con nome “Tachipirina” non esiste	Nome: Tachipirina	Il farmaco non esiste	–	PASS

7.2.5 EliminaFarmaco

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Nome del farmaco valido	Nome del farmaco valido	Il farmaco è presente nel catalogo	Nome: Tachipirina	Farmaco cancellato	Il farmaco viene cancellato dal catalogo	PASS
4	Il farmaco non esiste	Farmaco non presente nel sistema [ERROR]	Non esiste il farmaco chiamato “Tachipirina”	Nome: Tachipirina	Non puoi eliminare un farmaco che non esiste	–	PASS

7.2.6 ModificaFarmaco

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Tutti gli input validi, prescrizione True	Nome, Prezzo, Scorte e Prescrizione (True) validi	Il farmaco è presente nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: True	Modifica effettuata	Il farmaco viene correttamente modificato	PASS
2	Tutti gli input validi, prescrizione False	Nome, Prezzo, Scorte e Prescrizione (False) validi	Il farmaco è presente nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: False	Modifica effettuata	Il farmaco viene correttamente modificato	PASS
7	Il nome del farmaco non esiste	Nome non presente nel sistema [ERROR], Prezzo, Scorte e Prescrizione validi	Il farmaco “Tachipirina” non esiste nel sistema	Nome: Tachipirina Prezzo: 11.44 € Scorte: 100 Prescrizione: False	Il farmaco non esiste	–	PASS

7.2.7 GeneraReport

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Input	Output Attesi	Esito
1	Tutti gli input validi	DataInizio, DataFine validi	DataInizio: 01-06-2024 DataFine: 31-06-2024	Il report viene generato	PASS

7.2.8 GeneraOrdineAcquistoFarmacista

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Ordine valido	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort'	[('Tachipirina', 5), ('Fluifort', 10)]	Ordine di acquisto generato	Un ordine di acquisto viene correttamente creato	PASS
2	Ordine vuoto	Lista vuota [ERROR]	–	–	Non puoi creare un ordine di acquisto vuoto	–	PASS

7.2.9 RitiraOrdine

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Id dell'ordine valido	Id dell'ordine valido	–	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	Stato ordine cambiato con successo	Lo stato dell'ordine è stato cambiato con successo	PASS
4	Id inesistente	Id inesistente [ERROR]	Non esiste un farmaco con id "5ea930bc-f0a5-427a-8ca1-f9a2a6146948"	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	L'ordine non esiste	–	PASS

7.2.10 RegistraConsegnaOrdineAcquisto

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Id dell'ordine di acquisto valido	Id dell'ordine di acquisto valido	–	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	Ordine ricevuto	Lo stato dell'ordine di acquisto e le scorte in magazzino sono aggiornati	PASS
4	Id inesistente	Id inesistente [ERROR]	Non esiste un ordine di acquisto con Id "5ea930bc-f0a5-427a-8ca1-f9a2a6146948"	Id: 5ea930bc-f0a5-427a-8ca1-f9a2a6146948	L'ordine di acquisto non esiste	–	PASS

7.2.11 CreaOrdine

Test Case ID	Descrizione	Classi di Equivalenza Coperte	Pre-condizioni	Input	Output Attesi	Post-condizioni Attese	Esito
1	Ordine valido, l'ordine non esaurisce le scorte di nessun farmaco	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 5), ('Fluifort', 10)]	Ordine generato	Ordine creato: scorte in magazzino decrementate	PASS
2	Ordine valido, l'ordine esaurisce le scorte di un farmaco	Farmaci-Quantità valido	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 80), ('Fluifort', 10)]	Ordine generato	Ordine creato. Le scorte in magazzino vengono correttamente decrementate e viene generato un ordine di fornitura per il farmaco esaurito, richiedendone una quantità di default	PASS
3	Ordine vuoto	Lista vuota [ERROR]	–	–	Non puoi creare un ordine vuoto	–	PASS
4	Ordine invalido per scorte insufficienti	Scorte insufficienti [ERROR]	Esistono nel sistema i farmaci 'Tachipirina' e 'Fluifort' con scorte rispettivamente di 80 e 120	[('Tachipirina', 5), ('Fluifort', 150)]	Ordine non creato per scorte insufficienti	–	PASS