

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	4
2	Analisi e Specifica dei Requisiti	5
2.1	Analisi Nomi-Verbi	5
2.2	Revisione dei Requisiti	5
2.2.1	Requisiti Aggiuntivi	6
2.3	Glossario dei Termini	6
2.4	Classificazione dei Requisiti	7
2.4.1	Requisiti Funzionali	7
2.4.2	Requisiti sui Dati	7
2.4.3	Vincoli/Altri Requisiti	7
3	Modellazione dei Casi d'Uso	8
3.1	Attori e Casi d'Uso	8
3.2	Diagramma dei Casi d'Uso	9
3.3	Scenari	10
3.4	Diagramma delle Classi	15
3.5	Diagrammi di Sequenza	15
3.6	Verifica Completezza dei Requisiti	22
4	Piano di Test Funzionale	23
4.1	RegistraCliente	23
4.2	LoginUtente	25
4.3	AggiungiFarmaco	26
4.4	ModificaFarmaco	26
4.5	CercaFarmaco	28
4.6	EliminaFarmaco	28
4.7	GeneraReport	29
4.8	GeneraOrdineAcquistoFarmacista	29
4.9	RegistraConsegnaOrdineAcquisto	30
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
7	Testing	36
7.1	Test strutturale	36
7.1.1	creaOrdine()	36
7.1.2	modificaFarmaco()	39
7.2	Test funzionale	40
7.2.1	RegistraCliente	40
7.2.2	LoginUtente	41
7.2.3	AggiungiFarmaco	41
7.2.4	CercaFarmaco	41
7.2.5	EliminaFarmaco	41

7.2.6	ModificaFarmaco	42
7.2.7	GeneraReport	42
7.2.8	GeneraOrdineAcquistoFarmacista	42
7.2.9	RitiraOrdine	42
7.2.10	RegistraConsegnaOrdineAcquisto	43
7.2.11	CreaOrdine	43

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             if (!ordineAcquisto.getQuantitaFarmaci().isEmpty()) {
23                 ordineAcquisto.salvaInDB();
24             }
25             ordine.salvaInDB();
26             storicoOrdini.add(ordine);
27             return ordine.getId();
28         } catch (FarmacoNotFoundException e) {
29             throw new OrderCreationFailedException("Errore creazione ordine, farmaco non
30                 ↳ trovato");
31         } catch (DBException e) {
32             throw new OrderCreationFailedException(e.getMessage());
33         }
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 $R = \# \text{regioni chiuse in } G$

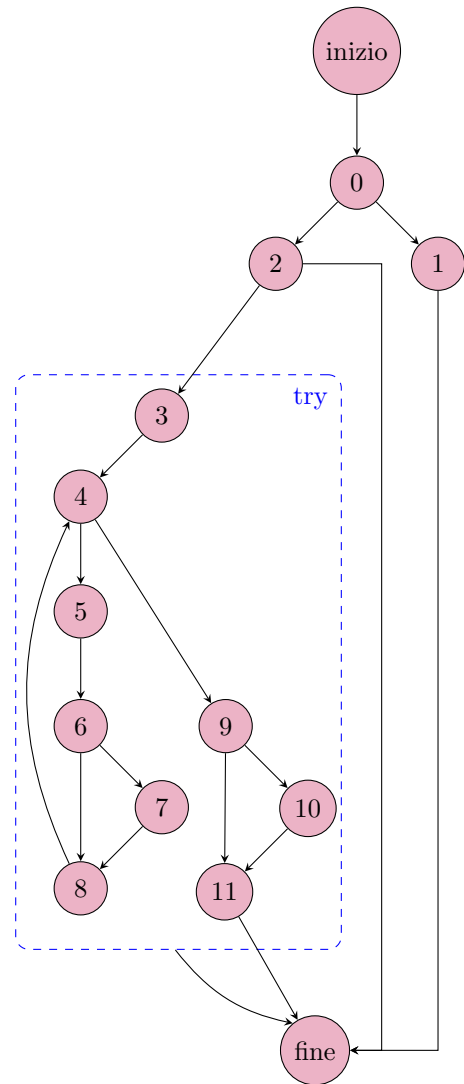
Nel nostro caso:

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

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

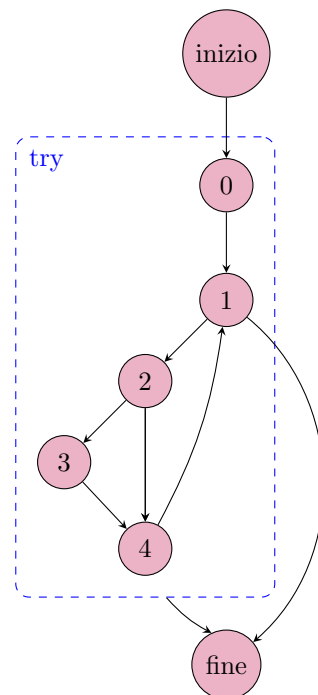
Nel nostro caso:

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

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 dell'applicazione della test suite definita nel Capitolo 4: Piano di Test Funzionale all'implementazione Java del progetto. Di seguito vengono quindi riportati gli esiti dell'esecuzione dei vari test case, in particolare non vengono mostrati quei casi di test basati esclusivamente sull'input validation, poiché la validazione del formato dei parametri di input viene effettuata completamente a livello Boundary. Tutti i test sono effettuati sul package `Controller`.

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: miorossi 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: mariorossi 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: mariorossi 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	PASS
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

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