



Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software*- Prof.ssa F.Ferrucci



# TP Test Plan

Chemio Plan

Riferimento	
Versione	1.0
Data	3/02/2020
Destinatario	Prof.ssa F. Ferrucci
Presentato da	
Approvato da	



## Revision History

Data	Versione	Cambiamenti	Autori
17/01/2020	1.0	Prima stesura	C. Fusco, V. Aiello



# Sommario

I.	<b>1. Introduzione.....</b>	<b>4</b>
II.	<b>2. Documenti correlati.....</b>	<b>4</b>
	2.1 Relazione con il documento di analisi.....	4
	2.2 Relazione con il System Design Document.....	4
	2.3 Relazione con l'Object Design Document .....	4
III.	<b>3. Panoramica del sistema .....</b>	<b>5</b>
IV.	<b>4. Funzionalità da testare .....</b>	<b>5</b>
V.	<b>5. Criteri Pass/Failed .....</b>	<b>6</b>
VI.	<b>6. Approccio.....</b>	<b>6</b>
	6.1 Testing di unità .....	6
	6.2 Testing di sistema .....	7
VII.	<b>7. Sospensione e ripresa .....</b>	<b>7</b>
	7.1 Criteri di sospensione .....	7
	7.2 Criteri di ripresa .....	7
	7.3 Criteri di terminazione.....	7
VIII.	<b>8. Materiale per il testing.....</b>	<b>Errore. Il segnalibro non è definito.</b>
IX.	<b>9. Test cases.....</b>	<b>8</b>
	9.1 Assistenza .....	<b>Errore. Il segnalibro non è definito.</b>
	9.2 Inventario .....	<b>Errore. Il segnalibro non è definito.</b>
	9.3 Cassa.....	<b>Errore. Il segnalibro non è definito.</b>
X.	<b>10. Riferimenti ad altri documenti di test .....</b>	<b>22</b>

## 1. Introduzione

---

Nell'affrontare alcuni aspetti fondamentali del sistema Chemio Plan ci siamo posti delle domande:

*Come ottenere un buon prodotto? Come essere sicuri che sia un buon prodotto?*

Nasce così la necessità di rilevare eventuali errori prodotti durante la fase di implementazione per evitare che essi si presentino nel momento in cui sistema verrà utilizzato dall'utente finale.

A tal proposito, in questo documento verranno definiti gli approcci e le attività di testing per verificare il corretto funzionamento del sistema sotto determinate condizioni.

Abbiamo pensato a opportuni casi e dati di input specifici in grado di mettere alla prova ogni singola funzionalità e caratteristica offerta dalla piattaforma.

I risultati dei test che verranno eseguiti saranno il punto cruciale nell'analisi delle failure e delle loro cause (fault) per individuare dove bisognerà intervenire per correggere gli errori o apportare modifiche per il miglioramento dei vari sottosistemi.

## 2. Documenti correlati

---

Questo documento è in stretta relazione con i documenti prodotti fino al rilascio della versione 1.0 del Test Plan e con i documenti che verranno sviluppati e rilasciati in futuro.

### 2.1 Relazione con il Requirement Analysis Document

La progettazione dei casi di test avviene prescindendo dalla conoscenza della struttura interna del prodotto ed operando solo sulle specifiche.

Per questo motivo facciamo riferimento al contenuto del documento di analisi che descrive dettagliatamente le funzionalità del sistema attraverso scenari, use case, diagrammi e mockup. Vi è inoltre indicata anche la priorità dei requisiti.

### 2.2 Relazione con il System Design Document

Nel system design document è stata definita la suddivisione in sottosistemi relativamente al prodotto che intendiamo presentare.

In questa fase è importante focalizzare la nostra attenzione sul layer di business. Verranno infatti testate tutte le funzionalità garantite nei sottosistemi specificati all'interno nel System Design Document relativamente al livello business.

### 2.3 Relazione con l'Object Design Document

Nell'Object Design Document (ancora non sviluppato al momento del rilascio dell' Test Plan v.1.0) sono contenute le classi che compongono il sistema e le loro mansioni. Faremo riferimento ad esse nel corso del documento per associare i test al codice prodotto.

### 3. Panoramica del sistema

---

ChemioPlan è una web app che fornisce un'interfaccia per le sue funzionalità: gestione dei pazienti, gestione dei farmaci, scheduling delle somministrazioni.

Il sistema che proponiamo prevede due attori principali:

- Dottore: potrà inserire, modificare e visualizzare tutte le prenotazioni e le schede paziente.
- Responsabile farmacia: potrà inserire stock di farmaci appena arrivati e potrà registrare anche un nuovo farmaco. Potrebbe altresì fare una richiesta di rifornimento facendosi consigliare da una previsione della quantità di prodotto da acquistare.

Nel System Design è stata definita l'architettura della piattaforma. In particolare, il livello di business è composto da sei sottosistemi:

- Utente: definisce l'utente (Dottore, responsabile farmacia) ed offre i relativi servizi di autenticazione.
- Prenotazione: modella tutto ciò che riguarda il processo di prenotazione somministrazioni.
- Paziente: modella tutte ciò che riguarda le schede paziente.
- Farmacia: modella la gestione dei farmaci e il relativo rifornimento.
- Ottimizzazione Scheduling: modella l'ottimizzazione delle prenotazioni.
- Ottimizzazione Acquisti: permette di ottimizzazione l'acquisto dei medicinali

### 4. Funzionalità da testare

---

La fase di testing avrà come obiettivo quello di testare i requisiti che presentano una priorità media o alta, di seguito l'elenco dei requisiti da testare per ogni gestione:

Il testing funzionale riguarderà nel dettaglio le funzionalità di seguito elencate (in base al sottosistema che le realizza):

- **Prenotazione**
  - Creazione nuova prenotazione
  - Modifica prenotazione
  - Visualizzazione prenotazione
- **Farmacia**
  - Inserimento lotto farmaci già registrato
  - Inserimento nuovo lotto farmaco
  - Modifica informazioni farmaco
  - Visualizzazione informazioni farmaco
  - Ordina farmaco
- **Paziente**
  - Visualizzazione informazioni paziente
  - Modifica informazioni paziente

- **Utente**
  - Login
  - Logout
- **Ottimizzazione Scheduling**
  - Suggerimento data prenotazione
- **Ottimizzazione Acquisti**
  - Suggerimento quantità farmaci da acquistare

## 5. Criteri Pass/Failed

---

Abbiamo determinato un insieme di input possibili che possano aiutarci a trovare errori nel sistema. Pertanto, il test ha successo se il comportamento osservato è diverso dal comportamento specificato nei requisiti funzionali.

Ogni qual volta verrà individuata una failure, legata ad un fault, analizzeremo i sottosistemi coinvolti nell'errore procedendo alla sua correzione. Una volta terminata la correzione verrà iterata la fase di testing per verificare che le modifiche apportate agli stessi non abbiano avuto impatti negativi su altre componenti del sistema. Il testing fallirà se gli non saranno trovati errori nelle componenti.

Presenza di errori: Pass, assenza di errori: Failed.

## 6. Approccio

---

La fase di testing si compone di tre attività: una prima fase si occuperà di trovare errori in una singola componente; la seconda fase, invece, avrà come compito quello di testare le funzionalità nate dall'integrazione dei vari sottosistemi e per ultimo andremo a testare l'intero sistema assemblato al fine di verificare soprattutto che esso soddisfi i desideri del cliente.

Di seguito verranno descritte brevemente le strategie individuate per effettuare il test di unità, d'integrazione, di sistema e di usabilità.

### 6.1 Testing di unità

Durante questa fase ci si focalizza sul comportamento di una componente permettendo di eseguire testing in modalità black-box o white-box.

Le nostre componenti saranno testate secondo il metodo white-box. Infatti, durante questa fase poseremo la nostra sulla struttura del codice che realizza le funzionalità fornite dalla componente al fine di individuare errori sia di logica che di implementazione.

## 6.2 Testing di sistema

La verifica sulle funzionalità del sistema avviene testando i possibili input degli utenti. La riduzione dei casi di test è attuata tramite l'adozione del category partition.

Il testing di sistema concluderà la fase di test del prodotto ed il primo ciclo di sviluppo. Per questa tipologia di test, ci affidiamo all'utilizzo di un software ausiliario come Katalon Studio al fine di osservare il comportamento del sistema in presenza di combinazioni di input utente non ammesse.

## 7. Sospensione e ripresa

---

Tenuto conto delle risorse necessarie impiegate durante la fase di testing, abbiamo stabilito dei criteri in base ai quali le attività di test saranno sospese o riprese.

### 7.1 Criteri di sospensione

Il test è sospeso se oltre il 10% dei casi di test falliscono. In queste condizioni, il team deve necessariamente correggere i fault prima di continuare implementazione e testing di nuove funzionalità.

### 7.2 Criteri di ripresa

Sarà necessario, dopo aver introdotto cambiamenti, testare le nuove componenti: se esse rivelano dei fault che impattano sulle componenti già esistenti, allora verranno testate di nuovo anche queste ultime.

I test case, quindi, verranno ancora una volta eseguiti per assicurarsi di aver risolto i nuovi problemi. In ogni caso, è nostra intenzione utilizzare sistemi di Continuous Integration per verificare in maniera automatizzata che ogni modifica fatta al codice non crei errori di integrazione con il codice preesistente.

### 7.3 Criteri di terminazione

Il test si considera terminato quando la totalità dei casi di test somministrati al sistema riporta esito negativo.

## 8. Materiale per il testing

---

L'esecuzione dei test necessita di un server correttamente configurato su cui siano installati Java e il DBMS. La configurazione deve avvenire come da manuale d'installazione.

Il testing è condotto utilizzando vari framework tra i più famosi ed efficaci in ambiente Web: JUnit e Katalon.

Come detto al punto precedente, i test sono eseguiti ad ogni modifica apportata al sistema, in un ambiente di Continuous Integration: ciò è possibile grazie all'utilizzo di Travis CI e Maven.

## 9. Test cases

Riferimento Use Case UC\_1

Test case per la funzionalità di inserimento di un nuovo lotto di farmaci.

Il responsabile della farmacia dovrà inserire:

- Codice Farmaco
- Quantità
- Nome del farmaco
- Numero lotto
- Scadenza

### VINCOLI

Parametro: Codice Farmaco	
Nome Categoria	Scelta per la Categoria
Formato [FC]	{0-z}{1,10} 1. Formato non corretto [ERRORE] 1. Formato corretto [PROPERTY_FC_OK]
Parametro: Quantità	
Nome Categoria	Scelta per la Categoria
Formato [FQ]	[0-9] 1. Formato non corretto [ERRORE] 1. Formato corretto [PROPERTY_FQ_OK]
Positività [PQ]	1. Quantità <= 0 [ERRORE] 2. Quantità > 0 [PROPERTY_PQ_OK]
Parametro: Nome Farmaco	
Nome Categoria	Scelta per la Categoria
Formato[FN]	[A-Za-z]{1-256} 1. Formato non corretto [ERRORE] 2. Formato corretto [PROPERTY_FN_OK]
Parametro: Numero lotto	
Nome Categoria	Scelta per la Categoria
Formato [FNL]	[0-9] 1. Formato non corretto [ERRORE] 2. Formato corretto [PROPERTY_FNL_OK]
Positività [PNL]	1. Numero lotto <= 0 [ERRORE] 2. Numero lotto > 0 [PROPERTY_PNL_OK]
Parametro: Scadenza	
Nome Categoria	Scelta per la Categoria





Correttezza [CS]	1. Scadenza < Data corrente [ERRORE] 2. Scadenza >= Data corrente [PROPERTY_CS_OK]
------------------	--

## TEST FRAMES

Test Case ID	Test frame	Esito
TC_1_1	FC1	Errato: Formato codice invalido
TC_1_2	FC2,FQ1	Errato: Formato quantità invalido
TC_1_3	FC2,FQ2,PQ1	Errato: Quantità negativa
TC_1_4	FC2,FQ2,PQ2,FN1	Errato: Formato nome farmaco invalido
TC_1_5	FC2,FQ2,PQ2,FN2,FNL1	Errato: Formato numero lotto invalido
TC_1_6	FC2,FQ2,PQ2,FN2,FNL2,PNL1	Errato: Numero lotto negativo
TC_1_7	FC2,FQ2,PQ2,FN2,FNL2,PNL2,CS1	Errato: Scadenza passata
TC_1_8	FC2,FQ2,PQ2,FN2,FNL2,PNL2,CS2	Corretto

## TEST CASES

Test Case ID	TC_1_1	Test Frame	FC1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
1. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	
Codice farmaco		FM11DFSJJFDKL	
Quantità		23	
Nome farmaco		Farmaco1	
Numero lotto		12	
Scadenza		1/01/2023	
2. Il responsabile della farmacia conferma e sottomette i dati			
Oracolo			
La modifica non va a buon fine in quanto il formato del codice non è corretto.			

Test Case ID	TC_1_2	Test Frame	FC2,FQ1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			

1. Il Responsabile Farmacia inserisce il codice del farmaco

Input	Valore
Codice farmaco	FM11
Quantità	AB
Nome farmaco	Farmaco1
Numero lotto	12
Scadenza	1/01/2023

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

La modifica non va a buon fine in quanto il formato della quantità non è corretto.

Test Case ID	TC_1_3	Test Frame	FC2,FQ2,PQ1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
3. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	
Codice farmaco		FM11	
Quantità		-23	
Nome farmaco		Farmaco1	
Numero lotto		12	
Scadenza		1/01/2023	
4. Il responsabile della farmacia conferma e sottomette i dati			
Oracolo			
La modifica non va a buon fine in quanto la quantità del farmaco è negativa.			

Test Case ID	TC_1_4	Test Frame	FC2,FQ2,PQ2,FN1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			

1. Il Responsabile Farmacia inserisce il codice del farmaco

Input	Valore
Codice farmaco	FM11
Quantità	23
Nome farmaco	
Numero lotto	12
Scadenza	1/01/2023

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

La modifica non va a buon fine in quanto il formato del nome non è corretto.

Test Case ID	TC_1_5	Test Frame	FC2,FQ2,PQ2,FN2,FNL1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
1. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	
Codice farmaco		FM11	
Quantità		23	
Nome farmaco		Farmaco1	
Numero lotto		sdfgn	
Scadenza		1/01/2023	
2. Il responsabile della farmacia conferma e sottomette i dati			
Oracolo			
La modifica non va a buon fine in quanto il formato del numero lotto non è corretto.			

Test Case ID	TC_1_6	Test Frame	FC2,FQ2,PQ2,FN2,FNL2,PNL1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			

1. Il Responsabile Farmacia inserisce il codice del farmaco

Input	Valore
Codice farmaco	FM11
Quantità	23
Nome farmaco	Farmaco1
Numero lotto	-12
Scadenza	1/01/2023

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

La modifica non va a buon fine in quanto il numero lotto è negativo.

Test Case ID	TC_1_7	Test Frame	FC2,FQ2,PQ2,FN2,FNL2,PNL2,CS1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
1. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	
Codice farmaco		FM11	
Quantità		23	
Nome farmaco		Farmaco1	
Numero lotto		12	
Scadenza		1/01/2000	
2. Il responsabile della farmacia conferma e sottomette i dati			
Oracolo			
La modifica non va a buon fine in quanto la scadenza è passata.			

Test Case ID	TC_1_8	Test Frame	FC2,FQ2,PQ2,FN2,FNL2,PNL2,CS2
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			

1. Il Responsabile Farmacia inserisce il codice del farmaco

Input	Valore
Codice farmaco	FM11
Quantità	23
Nome farmaco	Farmaco1
Numero lotto	12
Scadenza	1/01/2023

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

L'inserimento di un nuovo lotto di farmaci va a buon fine.

Riferimento Use Case UC\_2

Test case per la funzionalità di aggiornamento della quantità del farmaco.

Il responsabile della farmacia dovrà inserire:

- Codice Farmaco
- Quantità

## VINCOLI

Parametro: Codice Farmaco	
Nome Categoria	Scelta per la Categoria
Formato [FC]	[0-z]{1,10} 1. Formato non corretto [ERRORE] 2. Formato corretto [PROPERTY_FC_OK]
Parametro: Presenza nel Database	
Nome Categoria	Scelta per la Categoria
Presenza[PC]	3. Non presente nel DB [ERRORE] 4. Presente nel DB [PROPERTY_PC_OK]
Parametro: Quantità	
Nome Categoria	Scelta per la Categoria
Formato [FQ]	[0,9] 1. Formato non corretto [ERRORE] 2. Formato corretto [PROPERTY_FQ_OK]
Positività [PQ]	1. Quantità < 0 [ERRORE] 2. Quantità >= 0 [PROPERTY_PQ_OK]



## TEST FRAMES

Test Case ID	Test Case Frame	Esito
TC_2_1	FC1	Errato: Lunghezza codice farmaco errata
TC_2_2	FC2, PC1	Errore: Farmaco non presente nel DB
TC_2_3	FC2, PC2, FQ1	Errore: Formato non corretto
TC_2_4	FC2, PC2, FQ2, PQ1	Errore: Quantità farmaco negativa
TC_2_5	FC2, PC2, FQ2, PQ2	Corretto

## TEST CASES

Test Case ID	TC_2_3	Test Frame	FC2, PC2, FQ2, PQ1
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
3. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	
Codice Farmaco		774568CDS	
4. Il Responsabile Farmacia conferma l'inserimento			
5. Il Sistema mostra un form con tutti i dati del farmaco			
6. Il Responsabile Farmacia modifica la quantità del farmaco			
Input		Valore	
Quantità Farmaco		-50	
7. Il Responsabile Farmacia conferma il salvataggio delle modifiche			
Oracolo			
La modifica della quantità del farmaco non va a buon fine in quanto la quantità inserita è negativa			

Test Case ID	TC_2_4	Test Frame	FC2, PC2, FQ2, PQ2
Precondizione			
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco			
Flusso di Eventi			
1. Il Responsabile Farmacia inserisce il codice del farmaco			
Input		Valore	

Codice Farmaco	774568CDS
2. Il Responsabile Farmacia conferma l'inserimento 3. Il Sistema mostra un form con tutti i dati del farmaco 4. Il Responsabile Farmacia modifica la quantità del farmaco	
Input	Valore
Quantità Farmaco	75
5. Il Responsabile Farmacia conferma il salvataggio delle modifiche	
Oracolo	
La quantità del farmaco viene aggiornata	

Riferimento Use Case UC\_3

Test case per la funzionalità di ordine nel sottosistema Farmacia.

Il responsabile della farmacia dovrà inserire:

- Codice Farmaco
- Quantità

## VINCOLI

Parametro: Codice Farmaco	
Nome Categoria	Scelte per la categoria
Formato [FC]	[0-z] {1, 10} 1. Formato non corretto [errore] 2. Formato corretto [PROPERTY_FC_OK]
Parametro: Presenza Database	
Nome Categoria	Scelta per la categoria
Presenza DB [PC]	1. Non presente nel Database [errore] 2. Presente nel Database [PROPERTY_PC_OK]
Parametro: Quantità	
Nome Categoria	Scelta per la categoria
Formato [FQ]	[0-9] 1. Formato non corretto [errore] 2. Formato corretto [PROPERTY_FQ_OK]
Positività [PQ]	1. Quantità $\leq 0$ [errore] 2. Quantità $> 0$ [if PROPERTY_FQ_OK] [PROPERTY_PQ_OK]

## TEST FRAMES

Test Case ID	Test Frame	Esito
TC_3_1	FC1	Errato: formato codice farmaco errato
TC_3_1.1	FC2, PC1	Errato: codice non presente nel Database
TC_3_2	FC2, PC2, FQ1	Errato: formato quantità errato
TC_3_2.1	FC2, PC2, FQ2, PQ1	Errato: quantità non positiva
TC_3_3	LC2, PC2, FQ2, PQ2	Corretto

## TEST CASES

Test Case ID	TC_3_1	Test Frame	FC1
Precondizione			
Il responsabile di farmacia vuole ordinare un farmaco già registrato			
Flusso di Eventi			
1. Il responsabile inserisce i seguenti dati nel form:			
Input		Valore	
Codice Farmaco		ABC123456789	
Quantità		50	
2. Il Responsabile Farmacia conferma l'inserimento			
Oracolo			
L'ordine non va a buon fine in quanto la lunghezza del codice del farmaco è più di 10.			

Test Case ID	TC_3_3	Test Frame	FC2, PC2, FQ2, PQ2
Precondizione			
Il responsabile di farmacia vuole ordinare un farmaco già registrato			
Flusso di Eventi			



1. Il responsabile inserisce i seguenti dati nel form:

Input	Valore
Codice Farmaco	123456ABC
Quantità	50

2. Il Responsabile Farmacia conferma l'inserimento

Oracolo

L'ordine va a buon fine.

Riferimento Use Case UC\_4

Test case per la funzionalità di visualizzazione dei pazienti.

Il Dottore dovrà inserire, per la ricerca di un paziente:

- Codice Fiscale
- Nome
- Cognome
- Medicinale Utilizzato

## VINCOLI

Parametro: Codice Fiscale	
Nome Categoria	Scelte per la categoria
Formato [FP]	<p>[A-Z]{6}[0-9]{2}[A-Z]{1}[0-9]{2}[A-Z]{1}[0-9]{3}[A-Z]{1} ^.{0}</p> <p>1. Formato non Corretto [errore] 2. Formato Corretto [property FP_OK]</p>
Parametro: Nome	
Nome Categoria	Scelta per la categoria
Formato [FN]	<p>[A-z]{3,256} ^.{0}</p> <p>1. Formato non Corretto [errore] 2. Formato Corretto [property FN_OK]</p>
Parametro: Cognome	
Nome Categoria	Scelta per la categoria
Formato [FC]	<p>[A-z]{3,256} ^.{0}</p> <p>1. Formato non Corretto [errore]Formato Corretto [property FC_OK]</p>
Parametro: Codice Farmaco	



Nome Categoria	Scelte per la categoria
Formato [FF]	$[0-z]\{1,10\} \wedge.\{0\}$  1. Formato non Corretto [errore] 2. Formato Corretto [property FF_OK]

## TEST FRAMES

Test Case ID	Test Frame	Esito
TC_4_1	FP1	Errato: formato codice fiscale errato
TC_4_2	FP2,FN1	Errato: formato nome non errato
TC_4_3	FP2,FN2,FC1	Errato: formato cognome errato
TC_4_4	FP2,FN2,FC2,FF1	Errato: quantità codice farmaco errato
TC_4_5	FP2,FN2,FC2,FF2	Corretto

## TEST CASES

Test Case ID	TC_4_4	Test Frame	FP1
Pre-Condizione			
Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente			
Flusso di eventi			
1. Il Dottore inserisce i dati del farmaco nel form			
Input		Valore	
Codice Fiscale		JMSLBR30T84AKRNO	
Nome		Giorgio	
Cognome		Rossi	
Medicinale		12	
Codice Fiscale		JMSLBR30T84AKRNO	
2. Il Dottore conferma e sottomette i dati			
Oracolo			
L'evento non va a buon fine in quanto il formata del Codice fiscale non è corretto.			

Test Case ID	TC_4_5	Test Frame	FP2,FN2,FC2,FF2
Pre-Condizione			
Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente			
Flusso di eventi			
1. Il Dottore inserisce i dati del farmaco nel form			
Input		Valore	
Codice Fiscale		JMSLBR30T84A623O	
Nome		Giorgio	
Cognome		Rossi	
Medicinale		12	
Codice Fiscale		JMSLBR30T84AKRNO	
2. Il Dottore conferma e sottomette i dati			
Oracolo			
L'evento va a buon fine.			

Riferimento Use Case UC\_5

Test case per la funzionalità di modifica prenotazione paziente.

Il dottore per la modifica della prenotazione dovrà inserire:

- Codice prenotazione [Stringa 8 caratteri]
- Data e ora della nuova prenotazione

## VINCOLI

Parametro: Codice prenotazione	
Nome Categoria	Scelte per categoria
Formato [FC]	{0-z} {1,8} 1. .Formato non corretto [errore] 2. Formato corretto [Property_FC_OK]
Parametro: Presenza Database	
Nome Categoria	Scelta per categoria
Presenza [PC]	1. Codice prenotazione non presente nel database [Errore] 2. Codice prenotazione presente nel database [Property_PC_OK]
Parametro: Data e Ora	
Nome categoria	Scelta per categoria
Correttezza [CD]	1. DataOra <= DataOraCorrente [Errore] 2. .DataOra > DataOraCorrente [Property_CD_OK]



## TEST FRAMES

Test Case Id	Test frame	Esito
TC_5_1	FC1	Errato: formato codice prenotazione errato
TC_5_1.1	FC2, PC1	Errato: Codice prenotazione non presente nel DB
TC_5_2	FC2, PC2, CD1	Errato: Data nel passato
TC_5_3	FC2, PC2, CD2	Corretto

## TEST CASES

Test Case ID	TC_5_1	Test Frame	FC1
Pre-condizione			
Un dottore vuole modificare la prenotazione della somministrazione di un paziente			
Flusso di eventi			
1. Il dottore inserisce i seguenti dati nel form:			
Input		Valore	
Codice prenotazione		342vbnjnumu	
2. Il dottore conferma l'inserimento			
Oracolo			
La modifica della prenotazione non va a buon fine in quanto il codice prenotazione è troppo lungo e non rispetta il formato codice prenotazione [Stringa 8 caratteri]			

Test Case ID	TC_5_3	Test Frame	FC2, PC2, CD2
<b>Pre-condizione</b>			
Un dottore vuole modificare la prenotazione della somministrazione di un paziente			
<b>Flusso di eventi</b>			



Laurea Triennale in informatica-Università di Salerno  
Corso di *Ingegneria del Software*- Prof.ssa F.Ferrucci

1. Il dottore inserisce i seguenti dati nel form:

Input	Valore
Codice prenotazione	342543TV

2. Il dottore conferma l'inserimento  
3. Il sistema mostra la prenotazione relativa al codice e permette di modificarne data e ora

Input	Valore
Data e Ora	14/01/2023 10:00

4. Il dottore inserisce Data e ora della nuova prenotazione  
5. Il dottore Conferma l'inserimento

**Oracolo**

La modifica della prenotazione va a buon fine



## 10. Riferimenti ad altri documenti di test

---