



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
9/02/2023	2.0	Revisione Finale	[gruppo]



Sommario

I.	1. Introduzione	4
II.	2. Documenti correlati	4
	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.	3. Panoramica del sistema	5
IV.	4. Funzionalità da testare	5
V.	5. Criteri Pass/Failed	6
VI.	6. Approccio	6
	6.1 Testing di unità	6
	6.2 Testing di sistema	7
VII.	7. Sospensione e ripresa	7
	7.1 Criteri di sospensione.....	7
	7.2 Criteri di ripresa	7
	7.3 Criteri di terminazione	7
VIII.	8. Materiale per il testing	Errore. Il segnalibro non è definito.
IX.	9. Test cases	8
	9.1 Assistenza Errore. Il segnalibro non è definito.	
	9.2 Inventario Errore. Il segnalibro non è definito.	
	9.3 Cassa Errore. Il segnalibro non è definito.	
X.	10. Riferimenti ad altri documenti di test	27

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 porremo 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:

- Numero lotto
- Scadenza
- Quantità

VINCOLI

Parametro: Presenza nel Database	
Nome Categoria	Scelta per la Categoria
Presenza[PN]	1. Presente nel DB [ERRORE] 2. Non presente nel DB [PROPERTY_PN_OK]
Parametro: Scadenza	
Nome Categoria	Scelta per la Categoria
Correttezza [CS]	1. Scadenza < Data corrente [ERRORE] 2. Scadenza >= Data corrente [PROPERTY_CS_OK]
Parametro: Quantità	
Nome Categoria	Scelta per la Categoria
Positività [PQ]	1. Quantità < 0 [ERRORE] 1. Quantità >= 0 [PROPERTY_PQ_OK]

TEST FRAMES

Test Case ID	Test frame	Esito
TC_1_1	PN1	Errato: Presente nel Database
TC_1_2	PN2,CS1	Errato: Scadenza passata
TC_1_3	PN2,CS2,PQ1	Errato: Quantità negativa
TC_1_4	PN2,CS2,PQ2	Errato: Corretto



TEST CASES

Test Case ID	TC_1_1	Test Frame	PN1								
Precondizione											
Il Responsabile Farmacia vuole inserire un nuovo lotto di farmaci.											
Flusso di Eventi											
1. Il Responsabile Farmacia inserisce i seguenti dati nel form:											
<table><thead><tr><th>Input</th><th>Valore</th></tr></thead><tbody><tr><td>Numero lotto</td><td>2</td></tr><tr><td>Scadenza</td><td>22-02-2022</td></tr><tr><td>Quantità</td><td>12</td></tr></tbody></table>		Input	Valore	Numero lotto	2	Scadenza	22-02-2022	Quantità	12		
Input	Valore										
Numero lotto	2										
Scadenza	22-02-2022										
Quantità	12										
2. Il responsabile della farmacia conferma e sottomette i dati											
Oracolo											
L'inserimento di un nuovo lotto di farmaci non va a buon fine in quanto il lotto era già presente nel DataBase.											

Test Case ID	TC_1_2	Test Frame	PN2,CS1
--------------	--------	------------	---------



Precondizione

Il Responsabile Farmacia vuole inserire un nuovo lotto di farmaci.

Flusso di Eventi

1. Il Responsabile Farmacia inserisce i seguenti dati nel form:

Input	Valore
Numero lotto	12
Scadenza	22-01-2022
Quantità	12

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

L'inserimento di un nuovo lotto di farmaci non va a buon fine in quanto la data è nel passato.

Test Case ID

TC_1_3

Test Frame

PN2,CS2,PQ1

Precondizione

Il Responsabile Farmacia vuole inserire un nuovo lotto di farmaci.

Flusso di Eventi

1. Il Responsabile Farmacia inserisce i seguenti dati nel form:

Input	Valore
Numero lotto	12
Scadenza	22-01-2022
Quantità	-5

2. Il responsabile della farmacia conferma e sottomette i dati

Oracolo

L'inserimento di un nuovo lotto di farmaci non va a buon fine in quanto la quantità è negativa.

Test Case ID	TC_1_4	Test Frame	PN2,CS2,PQ2
Precondizione			
Il Responsabile Farmacia vuole inserire un nuovo lotto di farmaci.			
Flusso di Eventi			

1. Il Responsabile Farmacia inserisce i seguenti dati nel form:

Input	Valore
Numero lotto	12
Scadenza	22-01-2022
Quantità	23

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 di un lotto.

Il responsabile della farmacia dovrà inserire:

- Quantità

VINCOLI

Parametro: Quantità	
Nome Categoria	Scelta per la Categoria
Positività [PQ]	2. Quantità < 0 [ERRORE] 3. Quantità >= 0 [PROPERTY_PQ_OK]

TEST FRAMES



Test Case ID	Test Case Frame	Esito
TC_2_1	PQ1	Errato: Quantità negativa
TC_2_2	PQ2	Corretto

TEST CASES

Test Case ID	TC_2_1	Test Frame	PQ1						
Precondizione									
Il Responsabile Farmacia vuole aggiornare la quantità di un farmaco									
Flusso di Eventi									
1. Il Responsabile Farmacia inserisce il codice del farmaco									
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Scadenza</td><td>22-03-2022</td></tr><tr><td>Quantità</td><td>-5</td></tr></table>		Input	Valore	Scadenza	22-03-2022	Quantità	-5		
Input	Valore								
Scadenza	22-03-2022								
Quantità	-5								
2. Il Responsabile Farmacia conferma l'inserimento									
Oracolo									
La modifica della quantità del farmaco non va a buon fine in quanto la quantità è negativa.									

Test Case ID	TC_2_1	Test Frame	PQ1
--------------	--------	------------	-----

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
Scadenza	22-03-2022
Quantità	22
2. Il Responsabile Farmacia conferma l'inserimento	
Oracolo	
La modifica della quantità del farmaco va a buon fine.	

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: 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
Positività [PQ]	1. Quantità ≤ 0 [ERRORE] 2. Quantità > 0 [PROPERTY_PQ_OK]

TEST FRAMES

Test Case ID	Test Frame	Esito
TC_3_1	FC1	Errato: formato codice farmaco errato
TC_3_2	FC2, PQ1	Errato: quantità negativa
TC_3_3	FC2, 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		23	
Quantità		12	



2. Il responsabile conferma l'inserimento

Oracolo

L'ordine non va a buon fine in quanto il farmaco non è presente nel Database.

Test Case ID

TC_3_2

Test frame

FC2,FQ1

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	1
Quantità	-2

2. Il responsabile conferma l'inserimento

Oracolo

L'ordine non va a buon fine in quanto la quantità è negativa.



Test Case ID	TC_3_3	Test frame	FC2,FQ2						
Precondizione									
Il responsabile di farmacia vuole ordinare un farmaco già registrato									
Flusso di eventi									
1. Il responsabile inserisce i seguenti dati nel form:									
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Codice Farmaco</td><td>1</td></tr><tr><td>Quantità</td><td>13</td></tr></table>		Input	Valore	Codice Farmaco	1	Quantità	13		
Input	Valore								
Codice Farmaco	1								
Quantità	13								
2. Il responsabile 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

VINCOLI

Parametro: Codice Fiscale	
Nome Categoria	Scelte per la categoria
Formato [FP]	$[A-Z]\{6\}[0-9]\{2\}[A-Z]\{1\}[0-9]\{2\}[A-Z]\{1\}[0-9]\{3\}[A-Z]\{1\} \wedge.\{0\}$ <ol style="list-style-type: none"> 1. Formato non Corretto [errore] 2. Formato Corretto [property FP_OK]
Parametro: Nome	
Nome Categoria	Scelta per la categoria
Formato [FN]	$[A-z]\{3,256\} \wedge.\{0\}$ <ol style="list-style-type: none"> 1. Formato non Corretto [errore] 2. Formato Corretto [property FN_OK]
Parametro: Cognome	
Nome Categoria	Scelta per la categoria
Formato [FC]	$[A-z]\{3,256\} \wedge.\{0\}$ <ol style="list-style-type: none"> 1. Formato non Corretto [errore]Formato Corretto [property FC_OK]
Parametro: Codice Farmaco	
Nome Categoria	Scelte per la categoria
Formato [FF]	$[0-z]\{1,10\} \wedge.\{0\}$ <ol style="list-style-type: none"> 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	Corretto

TEST CASES



Test Case ID	TC_4_1	Test Frame	FP1								
Precondizioni											
Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente											
Flusso di Eventi											
1. Il Dottore Inserisce il codice fiscale per cercare un paziente.											
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Codice Fiscale</td><td>grgsrs 30T84A802O</td></tr><tr><td>Nome</td><td></td></tr><tr><td>Cognome</td><td></td></tr></table>				Input	Valore	Codice Fiscale	grgsrs 30T84A802O	Nome		Cognome	
Input	Valore										
Codice Fiscale	grgsrs 30T84A802O										
Nome											
Cognome											
2. Il Dottore Conferma l'inserimento.											
Oracolo											
La ricerca non va a buon fine in quanto il formato del codice fiscale è invalido.											

Test Case ID	TC_4_2	Test Frame	FP2,FN1
Precondizioni			
Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente			

Flusso di Eventi

1. Il Dottore Inserisce il codice fiscale per cercare un paziente.

Input	Valore
Codice Fiscale	
Nome	G
Cognome	

2. Il Dottore Conferma l'inserimento.

Oracolo

La ricerca non va a buon fine in quanto il formato del nome è invalido.

Test Case ID

TC_4_3

Test Frame

FP2,FN2,FC1

Precondizioni

Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente

Flusso di Eventi

1. Il Dottore Inserisce il codice fiscale per cercare un paziente.

Input	Valore



Codice Fiscale	
Nome	
Cognome	R

2. Il Dottore Conferma l'inserimento.

Oracolo

La ricerca non va a buon fine in quanto il formato del cognome è invalido.

Test Case ID

TC_4_4

Test Frame

FP2,FN2,FC2

Precondizioni

Un dottore si trova nella sezione Pazienti e vuole ricercare le informazioni di un paziente

Flusso di Eventi

1. Il Dottore Inserisce il codice fiscale per cercare un paziente.

Input	Valore
Codice Fiscale	
Nome	Giorgio
Cognome	



2. Il Dottore Conferma l'inserimento.

Oracolo

La ricerca 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:

- Data
- Sala
- Poltrona

VINCOLI

Parametro: Data e Ora	
Nome categoria	Scelta per categoria
Correttezza [CD]	1. DataOra <= DataOraCorrente [Errore] 2. .DataOra > DataOraCorrente [Property_CD_OK]
Parametro: Sala	
Nome Categoria	Scelte per categoria
Formato [FS]	[A-z]{3,256} ^.{0} 1. .Formato non corretto [errore] 2. Formato corretto [Property_FS_OK]
Parametro: Poltrona	
Nome Categoria	Scelta per categoria
Formato [FP]	[A-z]{3,256} ^.{0} 1. .Formato non corretto [errore] 2. Formato corretto [Property_FP_OK]



TEST FRAMES

Test Case Id	Test frame	Esito
TC_5_1	CD1	Errato: Data passata
TC_5_2	CD2.FS1	Errato: Formato sala invalido
TC_5_3	CD2.FS2,FP1	Errato: Formato poltrona invalido
TC_5_4	CD2.FS2,FP2	Corretto

TEST CASES

Test Case ID	TC_5_1	Test Frame	CD1								
Precondizione											
Un dottore vuole modificare la prenotazione della somministrazione di un paziente											
Flusso di eventi											
1.Il dottore inserisce i seguenti dati nel form:											
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Data</td><td>22-01-2023</td></tr><tr><td>Sala</td><td>Sala1</td></tr><tr><td>Poltrona</td><td>Poltrona1</td></tr></table>		Input	Valore	Data	22-01-2023	Sala	Sala1	Poltrona	Poltrona1		
Input	Valore										
Data	22-01-2023										
Sala	Sala1										
Poltrona	Poltrona1										
2.Il dottore conferma l'inserimento											
Oracolo											



La modifica della prenotazione non va a buon fine in quanto la data è nel passato.

Test Case ID	TC_5_2	Test Frame	CD2,FS1								
Precondizione											
Un dottore vuole modificare la prenotazione della somministrazione di un paziente											
Flusso di eventi											
1.Il dottore inserisce i seguenti dati nel form:											
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Data</td><td>22-01-2023</td></tr><tr><td>Sala</td><td>S</td></tr><tr><td>Poltrona</td><td>Poltrona1</td></tr></table>		Input	Valore	Data	22-01-2023	Sala	S	Poltrona	Poltrona1		
Input	Valore										
Data	22-01-2023										
Sala	S										
Poltrona	Poltrona1										
2.Il dottore conferma l'inserimento											
Oracolo											
La modifica della prenotazione non va a buon fine in quanto il formato della sala è invalido.											



Test Case ID	TC_5_3	Test Frame	CD2,FS2,FP1
Precondizione			
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	
Data		22-01-2023	
Sala		Sala1	
Poltrona		P	
2.Il dottore conferma l'inserimento			
Oracolo			
La modifica della prenotazione non va a buon fine in quanto il formato della poltrona è invalido			



Test Case ID	TC_5_4	Test Frame	CD2,FS2.FP2								
Precondizione											
Un dottore vuole modificare la prenotazione della somministrazione di un paziente											
Flusso di eventi											
1.Il dottore inserisce i seguenti dati nel form:											
<table><tr><th>Input</th><th>Valore</th></tr><tr><td>Data</td><td>22-02-2023</td></tr><tr><td>Sala</td><td>Sala1</td></tr><tr><td>Poltrona</td><td>Poltrona1</td></tr></table>		Input	Valore	Data	22-02-2023	Sala	Sala1	Poltrona	Poltrona1		
Input	Valore										
Data	22-02-2023										
Sala	Sala1										
Poltrona	Poltrona1										
2.Il dottore conferma l'inserimento											
Oracolo											
La modifica della prenotazione va a buon fine											



10. Riferimenti ad altri documenti di test
