

# Test Plan

---

*Vanessa Scovotto*

*Vincenzo Esposito*

*Gino Farisano*

*Giuseppe Valitutto*

**Partecipanti:**

<b>Nome</b>	<b>Matricola</b>
Vanessa Scovotto	0522500250
Giuseppe Valitutto	0522500267

## Indice

1.Introduzione .....	4
2.Panoramica del sistema .....	4
3. Funzionalità da testare/non testare .....	5
4.Criteri Pass/Failed .....	5
5. Approccio .....	6
6.Sospensione e ripristino .....	7
7.Materiale per il testing .....	7
8.Casi di test.....	7
9.Specifica dei test cases .....	7
9.1 Registra Nuovo Paziente .....	8
9.2Modifica paziente.....	11
9.3Ricerca Paziente .....	13
9.4Cancellazione Appuntamento .....	14
9.5Ricerca appuntamento .....	14
9.6Inserimento Appuntamento.....	15
9.7 Login .....	16
9.8 Inserimento Appuntamento .....	18
9.9 Visualizza Appuntamento .....	19
10.Pianificazione del testing.....	20
11.Schedulazione delle attività.....	20

## 1.Introduzione

Il Test Plan mostra in che modo viene gestita la fase di testing del sistema. Esso presenta tutte le attività di testing adottate, la quantità di tempo che viene usato per eseguire l'attività di testing, gli strumenti usati per testare e le componenti da testare. Il testing è una tecnica di “fault detection” il cui scopo è appunto quello di rilevare errori in maniera pianificata all'interno del codice prodotto dal mapping. L'obiettivo è quello di evitare che essi si presentino in fase di utilizzo all'utente finale. Per questo motivo un test che ha successo è un test che ha identificato un errore, non un test che “garantisce” la correttezza del codice. Lo scopo del presente documento è quello di descrivere e pianificare l'attività di testing del sistema Full Dent.

In esso vengono pianificate le attività di test relative a:

- gestione paziente
- gestione appuntamento

Si noti, tuttavia, che verranno testate esclusivamente alcune delle funzionalità implementate. Nel documento, oltre alla gestione dei test delle funzionalità, vengono anche pianificate le responsabilità dello staff e lo scheduling del test. La fase di testing è strettamente legata alle fasi ad essa precedenti; ogni documento, risultato delle differenti fasi di sviluppo, sarà punto di partenza indispensabile per poter effettuare un testing corretto e adeguato.

## 2.Panaromica del sistema

Il sistema Full Dent, utilizza un'architettura three-tier che divide il sistema in tre livelli:

- Presentation layer
- Business logic layer
- Services layer

Il primo livello descrive le interfacce con cui interagiscono gli utenti del sistema. Il secondo livello si occupa della logica funzionale del sistema, mentre l'ultimo livello si occupa della gestione dei dati persistenti. La scelta dei test da effettuare sarà molto accurata e soprattutto partirà dal presupposto che il sistema sicuramente presenta degli errori e l'obiettivo dei test sarà proprio di scovare tali errori per poi poterli risolvere.

### 3. Funzionalità da testare/non testare

L'attività di testing comprenderà tutte le funzionalità di cui dispone il nuovo sistema, quindi tutte le funzionalità ereditate dal vecchio sistema, con le opportune modifiche implementative dove è risultato necessario, e tutte le nuove funzionalità offerte e opportunamente implementate. Di seguito sono elencate tutte le funzionalità offerte dal sistema.

Area	Funzionalità
Gestione paziente	<ul style="list-style-type: none"><li>• Registrazione Nuovo Paziente</li><li>• Login</li><li>• Modifica dati di un paziente</li><li>• Ricerca Paziente</li></ul>
Gestione Appuntamento	<ul style="list-style-type: none"><li>• Inserimento Appuntamento</li><li>• Elimina appuntamento</li><li>• Ricerca Appuntamento</li></ul>

### 4.Criteri Pass/Failed

#### ➤ Criteri di successo

Nel caso in cui il risultato del testing sia un insuccesso, cioè che il comportamento offerto dal sistema coincida con la volontà del cliente e che non si verifichino errori tecnici, si può affermare che il sistema svolge il compito per cui è stato progettato nelle modalità previste e quindi il successo della componente o dell'insieme dei componenti testati. I criteri di successo si possono classificare in due categorie:

- **Comportamento Atteso**  
Le varie componenti svolgono il lavoro previsto.
- **Nessun errore rilevato dalle componenti**  
La componente non genera alcuna eccezione non prevista.

#### ➤ Criteri di insuccesso

Nel caso il cui risultato del testing sia un successo, cioè che il comportamento offerto dal sistema non coincida con la volontà del cliente e che quindi si verifichino errori, si può affermare che il sistema non svolge il compito per cui è stato progettato nelle modalità previste e quindi il relativo insuccesso della

componente o dell'insieme dei componenti testati. I criteri di insuccesso si possono classificare in due categorie:

- **Comportamento non atteso**  
Le varie componenti non svolgono il lavoro previsto
- **Errore rilevato dalle componenti**  
La componente genera eccezioni non previste

## 5. Approccio

Nella sessione di testing di FullDent verrà utilizzato un approccio di tipo "BLACK BOX", che prevede che i testing vengano effettuati in maniera da non scendere nei dettagli del codice, ma basandosi sulle specifiche delle funzionalità da testare.

L'approccio si compone in 3 fasi:

- Testing di unità, che controlla i singoli componenti( classi, metodi)
- Testing di integrazione, che va a testare l'integrazione dei vari sottosistemi
- Testing di sistema: test funzionale, che andrà a verificare la funzionalità dell'intero sistema assemblato.

### Testing di unità

Con il testing di unità verrà effettuato un controllo delle varie classi e metodi del sistema, quindi saranno ricercate le condizioni di fallimento andando ad evidenziare gli errori. Il testing di unità, sarà eseguito dal team di sviluppo attraverso l'implementazione di classi di test utilizzando il framework JUnit. In particolare, per ogni classe che esegue operazioni complesse sarà sviluppata la relativa classe JUnit.

### Testing di integrazione

Con il testing di integrazione si effettuerà un controllo sull'integrazione delle varie componenti del sistema. Si adotterà una strategia di tipo "Bottom-up". La lista dei casi di test verrà fuori dall'applicazione del metodo del Category Partition .

### Testing di sistema

Con il testing di sistema verrà effettuato un controllo della correttezza

dell'intero sistema. E' da considerare il testing più critico, in quanto può risultare molto complesso andare alla ricerca di eventuali errori, essendo impegnati tutti i sottosistemi. Questo test sarà effettuato manualmente.

## 6.Sospensione e ripristino

### ➤ Criteri di sospensione

Il testing sarà sospeso quando saranno state testate tutte le classi scelte per ogni possibile input e il risultato sarà quello atteso o sperato. Nel momento in cui il testing rivelerà un errore si dovrà passare alla fase di correzione che dovrà essere seguita da una ripetizione dell'intero testing per rilevare la presenza di errori introdotti dalle correzioni stesse.

### ➤ Criterio di ripristino

La ripresa in seguito alla fase di correzione avverrà, come già accennato, ripetendo l'intero testing per rilevare errori introdotti dalle correzioni.

## 7.Materiale per il testing

L'hardware necessario è un computer e uno smartphone. I software necessari sono la JDK, un web server (Tomcat) , eclipse e un DBMS (es. MySQL).

## 8.Casi di test

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition. Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

I test cases verranno definiti nel documento di Test Cases Specification (TCS).

## 9.Specifica dei test cases

Il nostro sistema lavora con due client, elencheremo prima i test cases relativi al client "segretaria".

## 9.1 Registra Nuovo Paziente

<b>Parametro</b>	
<b>Codice Fiscale</b>	
Formato[a-z][a-z][a-z][a-z][a-z][a-z][0-9][0-9][a-z][0-9][0-9][a-z][0-9][0-9][a-z]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza cf	1: lunghezza <15[errore]
	2:lunghezza=15 [ok]
	3:lunghezza >15 [error]
Formato ff	1: rispecchia il formato [if lunghezzacfok] [property formatoffok, rispecchia il formato]
	2: non rispetta il formato [if lunghezzacfok][errore]
<b>Parametro nome</b>	
Formato:[A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza ln	1: lunghezza<2 [errore]
	2:lunghezza >2[property lunghezzalnok]
Formato fn	1: rispetta il formato [if lunghezzalnok] [property formatofnok, rispetta il formato [A-Za-z]
	2. non rispetta il formato [if lunghezzalnok][error]



<b>Parametro Cognome</b>	
Formato[A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza lc	1: lunghezza <2 [error] 2: lunghezza >2 [property lunghezzalcok]
Formato fc	1:rispecchia il formato [if lunghezzalcok][property formatofcok, rispecchia il formato [A-za-z] 2:non rispetta il formato [if lunghezzalcok][error]
<b>Parametro Indirizzo</b>	
Formato: Via [a-z]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza li	1: lunghezza <7 [error] 2: lunghezza[7-20] [property lunghezzaliok]
Formato fi	1: respecchia il formato [if lunghezzaliok][property formatofiok, rispecchia il formato: Via[a-z] 2: non respecchia il formato [lunghezzaliok][error]
<b>Parametro Telefono</b>	
Formato[0-9]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza lt	1: lunghezza <10 [error] 2: lunghezza >10 [error]

	3: lunghezza=10 [property lunghezzaltok]
Formato ft	1: rispecchia il formato: [if lunghezzaltok][property formatoftok, rispecchia il formato[0-9]  2: non rispecchia il formato [if lunghezzaltok][error]
<b>Parametro email</b>	
Formato[A-Za-z0-9._]@[a-z.]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza le	1: lunghezza <10[error]  2: lunghezza >=10 [property lunghezzaleok]
Formato fe	1: rispecchia il formato [if lunghezzaleok][property formatofoek, rispecchia il formato [A-Za-z0-9. ]@[a-z.]]  2: non rispetta il formato [if lunghezzaleok][error]

<b>Codice</b>	<b>Combinazione</b>	<b>Esito</b>
TC 1.01	Cf1	error
TC 1.02	Cf3	error
Tc1.03	cf2ff2	error
TC1.04	Cf2ff1ln1	error
TC1.05	Cf2ff1ln2fn2	error

TC1.06	Cf2ff1ln2fn1lc1	error
TC1.07	Cf2ff1ln2fn1lc2fc2	error
TC1.08	Cf2ff1ln2fn1lc2fc1li1	error
TC1.09	Cf2ff1ln2fn1lc2fc1li2fi2	error
TC1.10	Cf2ff1ln2fn1lc2fc1li2fi1lt1	error
Tc 1.11	Cf2ff1ln2fn1lc2fc1li2fi1lt2ft2	error
Tc1.12	Cf2ff1ln2fn1lc2fc1li2fi1lt2ft1le1	error
Tc1.13	Cf2ff1ln2fn1lc2fc1li2fi1lt2ft1le2fe2	error
Tc1.14	Cf2ff1ln2fn1lc2fc1li2fi1lt2ft1le2fe1	inserimento

## 9.2 Modifica paziente

<b>Parametro indirizzo</b>	
Formato: Via[a-z]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza li	1: lunghezza <7 [error] 2: lunghezza[7-20] [property lunghezzaliok]
Formato fi	1: rispecchia il formato [if lunghezzaliok][property formatofiok, rispecchia il formato: Via[a-z] 2: non rispecchia il formato [lunghezzaliok][error]
<b>Parametro Telefono</b>	
Formato [0-9]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza lt	1: lunghezza <10 [error]

	2: lunghezza=10 [property lunghezzaltok]
Formato ft	1: rispecchia il formato: [if lunghezzaltok][property formatoftok, rispecchia il formato[0-9]  2: non rispecchia il formato [if lunghezzaltok][error]
<b>Paremtro email</b>	
Formato[A-Za- z0-9._]@[a-z.]	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza le	1: lunghezza <8[error]  2: lunghezza >=8 [property lunghezzaleok]
Formato fe	1: rispecchia il formato [if lunghezzaleok][property formatofoek, rispecchia il formato [A-Za-z0-9. ]@[a-z.]]  2: non rispetta il formato [if lunghezzaleok][error]

<b>codice</b>	<b>combinazione</b>	<b>esito</b>
TC 2.01	Li1	error
TC 2.02	Li2fi2	error
Tc2.03	Li2fi1lt1	error
TC2.04	Li2fi1lt1ft2	error

TC2.05	Li2fi1lt1ft1le1	error
TC2.06	Li2fi1lt1ft1le2fe1	modifica

### 9.3 Ricerca Paziente

<b>Parametro Cognome</b>	
Formato:[A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
Esiste ec	1: non esiste nel db [error] 2:esiste nel db [if lunghezzalcok and formatofcok]
<b>Parametro Nome</b>	
Formato [A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
Esiste en	1: non esiste nel db [error] 2:esiste nel db [if lunghezzalcok and formatofcok]

codice	combinazione	esito
TC 3.01	Ec1	error
TC 3.02	Ec2	Paziente Rilevato.
TC.3.03	En1	error
TC 3.03	En2	Paziente rilevato

#### 9.4 Cancellazione Appuntamento

Parametro Appuntamento	
<b>Categoria</b>	<b>Scelta</b>
Cancella ca	1: cancella[property cancellacaok]

codice	combinazione	esito
TC 4.01	Ca1	cancellazione

#### 9.5 Ricerca appuntamento

Parametro Data	
Formato [1-31]-[1-12]-[2013/2014]	
<b>Categoria</b>	<b>Scelta</b>
Formato fd	1: rispetta il formato[property formatofdok, rispetta il formato [1-31]-[1-12]-[2013/2014] 2. non rispetta il formato[error]
Prenotabile pd	1: sono giorni festivi [error] 2: è un giorno lavorativo

codice	combinazione	esito
TC5.01	Fd2	error
TC 5.02	Fd1pd1	error
TC 5.03	Fd2pd2	error
TC 5.04	Fd1pd2	Appuntamento rilevato.

## 9.6 Inserimento Appuntamento

<b>Parametro Data</b>	
Formato [1-31]-[1-12]-[2013/2014]	
<b>Categoria</b>	<b>Scelta</b>
Formato fd	1: rispetta il formato[property formatofdok, rispetta il formato [1-31]-[1-12]-[2013/2014]  2. non rispetta il formato[error]
Prenotabile pd	1: sono giorni festivi [error]  2: è un giorno lavorativo
<b>Parametro Servizio</b>	
Formato[a-z]	
<b>Categoria</b>	<b>Scelte</b>
Esiste es	1: non esiste nel db [error]  2:esiste nel db [if lunghezzalsok and formatofsok]
<b>Parametro Cliente</b>	
Formato[A-Z][a-z] [A-Z][a-z]	
<b>Categoria</b>	<b>Scelte</b>
Esiste ec	1: non esiste nel db [error]  2:esiste nel db [if lunghezzalcok and formatofcok]
<b>Parametro Ora</b>	
<b>Categorie</b>	<b>Scelte</b>

Giàoccupata go	1: ora già occupata[error] 2:ora libera
<b>Parametro Dottore</b>	
<b>Categoria</b>	<b>Scelte</b>
Giàoccupato gd	1: dottore già impegnato[error] 2: dottore libero

codice	combinazione	esito
TC6.01	Fd2	error
TC6.02	Fd1pd1	error
TC6.03	Fd2pd2es1	error
TC6.04	Fd2ed2es2ec1	error
TC6.05	Fd2ed2es2ec2go1	error
TC6.06	Fd2ed2es2ec2go2gd1	error
TC6.07	Fd2ed2es2ec2go2gd2	Inserimento Corretto.

Ora verranno elencati le specifiche dei test cases relativi al client “Utente” che utilizza l’applicazione dal proprio smartphone.

### 9.7 Login

<b>Parametro Password</b>	
<b>Categorie</b>	<b>Scelte</b>
Lunghezza lp	1: lunghezza <4[error] 2: lunghezza 5-10[property]



	lunghezzalpok] 3: lunghezza >10 [error]
Esiste ep	1: non esiste nel db [error] 2:esiste nel db [if lunghezzalpok]
Parametro UserName	
Categorie	Scelte
Lunghezza lu	1: lunghezza <4[error] 2: lunghezza 5-10[property lunghezzaluok] 3: lunghezza >10 [error]
Esiste en	1: non esiste nel db [error] 2:esiste nel db [if lunghezzaluok]

<b>codice</b>	<b>combinazione</b>	<b>esito</b>
TC7.01	Lp1	error
TC7.02	Lp3	error
TC7.03	Lp2ep1	error
TC7.04	Lp2ep2lu1	error
TC7.05	Lp2ep2lu2en1	error
TC7.06	Lp2ep2lu2en2	Autenticazione.

## 9.8 Inserimento Appuntamento

<b>Parametro Ora</b>	
<b>Categorie</b>	<b>Scelte</b>
Giàoccupata go	1: ora già occupata[error] 2:ora libera
<b>Parametro Dottore</b>	
<b>Categorie</b>	<b>Scelte</b>
Occupato do	1: dottore già impegnato[error] 2: dottore libero
<b>Parametro Servizio</b>	
<b>Categorie</b>	<b>Scelte</b>
Esiste es	1: non esiste nel db [error] 2:esiste nel db

<b>codice</b>	<b>combinazione</b>	<b>esito</b>
TC8.01	Go1	error
TC8.02	Go2do1	error
TC8.03	Go2do2ls1	error
TC8.04	Go2do2ls2es1	error
TC8.05	Go2do2ls3es2	Inserimento corretto

## 9.9 Visualizza Appuntamenti

<b>Parametro Data1</b>	
Formato[1-31]-[1-12][2013 oppure 2014]	
<b>Categorie</b>	<b>Scelte</b>
Formato fd	<p>1: rispetta il formato[property formatofdok, rispetta il formato [1-31]-[1-12]-[2013/2014]</p> <p>2. non rispetta il formato[error]</p>
<b>Parametro data2</b>	
Formato[1-31]-[1-12][2013 oppure 2014]	
<b>Categorie</b>	<b>Scelte</b>
Formato fdd	<p>1: rispetta il formato[property formatofdok, rispetta il formato [1-31]-[1-12]-[2013/2014]</p> <p>2. non rispetta il formato[error]</p>
Antecedenza ad	<p>1:la data deve essere antecedente alla prima data inserita</p> <p>2: la data è precedente alla prima data inserita[error]</p>

codice	combinazione	esito
TC9.01	Fd2	error
TC9.02	Fd1fdd2	error
TC9.03	Fd1fdd1ad2	error
TC9.04	Fd1fdd1ad1	Ricerca effettuata

**N.B** Il test case relativo alla cancellazione di un appuntamento è uguale a quello della segretaria.

## **10.Pianificazione del testing**

Il team per il testing deve essere composto da persone che hanno una completa e chiara conoscenza del sistema e delle tecniche di testing. Tali tecniche devono essere applicate nei tempi, nel budget e nei vincoli di qualità stabiliti.

Solitamente i componenti del team di testing non sono coinvolti nella fase di sviluppo, in quanto non deve esistere una reticenza nel decomporre il sistema. Il team dedicato al controllo della qualità è responsabile dell'attività di testing e quindi alla ricerca di fault. La documentazione dei fault è inviata agli sviluppatori per consentire la correzione del sistema. Il sistema revisionato poi è testato di nuovo non solo per verificare se gli errori trovati in precedenza sono stati corretti ma soprattutto per verificare che non si sono introdotti nuovi errori. L'attività di testing è molto importante in quanto la mancanza di tale attività o una cattiva interpretazione di essa può portare al completo fallimento del sistema.

### **➤ Determinazione dei ruoli**

Il team dedicato all'attività di testing di sistema sarà composto da Giuseppe Valitutto, Vanessa Scovotto, Esposito Vincenzo. Le attività relative al testing di unità saranno delegate agli stessi sviluppatori delle componenti, in modo tale da alleggerire il lavoro del team di testing.

## **11.Schedulazione delle attività**

Si prevede un tempo di 20 giorni per effettuare tutti i test.