



# Test Plan



Riferimento	
Versione	0.1
Data	15/10/2021
Destinatario	Gravino Carmine
Presentato da	S.Cataldo, P.Guidotti, C.Fiumarella, C.Orrigo



Approvato da

Nome e Cognome	Matricola
Onelia Sara Cataldo	0512105504
Paolo Guidotti	0512105261
Chiara Orrigo	0512105090
Camilla Fiumarella	0512105213

## Cronologia revisioni

Data	Versione	Descrizione	Autori
26/11/2021	0.1	Inizio stesura capitoli 1) Introduzione 2) Relazioni con gli altri documenti	Camilla Fiumarella, Paolo Guidotti
27/11/2021	0.2	Aggiunta capitoli: 3) Panoramica del sistema 4) Funzionalità da testare	Chiara Orrigo, Sara Cataldo
01/12/2021	0.3	Aggiunta capitoli: 5) Pass/Fail criteria 6) Approccio 7) Strumenti per il testing (hardware/software) 8) Test cases	[Tutti]



Laurea Magistrale in informatica-Università di Salerno Corso di  
*Gestione dei Progetti Software*- Prof. C.Gravino

05/12/2021	0.4	Aggiunta paragrafi: 6.1, 6.2, 6.3	Chiara Orrigo Sara Cataldo
15/12/2021	0.5	Aggiunta del capitolo: 9) Specifica dei test cases e paragrafo 9.1	[Tutti]
22/12/2021	0.6	Revisione paragrafo 9.1	Sara Cataldo Paolo Guidotti
07/12/2021	0.7	Aggiunta capitolo: 10) Pianificazione del testing e Assegnazione dei ruoli	Chiara Orrigo Camilla Fiumarella
10/02/2022	0.8	Revisione intero documento	[Tutti]



# Indice

## **1.Introduzione**

## **2.Relazione con gli altri documenti**

## **3.Panoramica del sistema**

## **4.Funzionalità da testare e non**

## **5.Pass/Fail criteria**

## **6.Approccio**

## **7.Strumenti per il testing (hardware/software)**

## **8.Test cases**

## **9.Specifica dei test cases**

### 9.1 Gestione cicli di dottorato

#### TC\_1.1 Gestione Autenticazione

##### 1.1.2 Login

#### TC\_1.2 Gestione Dipendente

##### 1.2.1 Inserimento Dipendente

## **10.Pianificazione del testing**



## 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, gli strumenti usati per testare e le componenti da testare, la quantità di tempo che viene usato per eseguire l'attività di testing. La tecnica del testing viene utilizzata per la ricerca e l'identificazione di fallimenti del sistema quindi la fase di testing ha esito positivo quando vengono trovati bug durante la diagnosi. Quando si verifica un comportamento anomalo del sistema, rispetto a quello previsto, attraverso lo studio della failure vengono identificati i faults dagli sviluppatori e viene eseguita la modifica e correzione. Il testing verrà svolto per le sole funzionalità implementate.

## 2. Relazione con gli altri documenti

Per verificare la corretta implementazione delle varie componenti del sistema sono stati predisposti dei test cases basati sui requisiti funzionali identificati nella fase di analisi e nella specifica dei requisiti. I documenti ai quali si fa riferimento:

- Relazioni con il Test Plan (TP) ;
- Relazioni con il Requirements Analysis Document(RAD);
- Relazioni con il System Design Document (SDD);
- Relazioni con il Object Design Document (ODD);

## 3. Panoramica del sistema

Il suo sviluppo è stato concepito in maniera modulare e pertanto è stato possibile dividerlo in sottosistemi (si veda il System Design Document(SDD)); Questo approccio, apporta vantaggi dal punto di vista della manutenzione e sarà fondamentale nella fase di testing, in quanto si potrà individuare un errore nel singolo sottosistema effettuando un numero limitato di test.

## 4. Funzionalità da testare e non

La lista delle features di cui si effettuerà il testing per le varie gestioni :

- **Gestione Registrazione**



- Registrazione Cliente
- Login
- Inserimento Dipendente
- **Gestione Dipendente**
  - Richiedi rifornimento
- **Gestione Ricambi**
  - Aggiorna ricambio

## 5. Pass/Fail criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individuerà un fallimento. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

## 6. Approccio

La fase di testing si suddivide in tre fasi principali che andremo ad eseguire per verificare il corretto funzionamento del nostro sistema :

L'approccio alla fase di testing si compone di tre 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.

### 6.1 Testing di unità

Per il testing di unità la metodologia consiste nel testare ogni metodo delle classi del sistema. Da esse, sono, quindi, escluse le interfacce e le classi entity, poiché quest'ultime presentano solo metodi getters e setters.

I casi di test verranno definiti mediante un approccio black-box e verranno documentati all'interno nel codice, attraverso il framework per il testing di classi Java JUnit.

Altre tecnologie usate in tale fase saranno:



- Mockito: per la costruzione degli stub e l'isolamento della componente testata.
- Maven: per la build e l'esecuzione automatica dei tests.

## 6.2 Testing d'integrazione

Con il testing d'integrazione si effettuerà un controllo sull'integrazione delle varie componenti del sistema. Si adotterà una strategia di tipo "Bottom-up". Per effettuare questi test di integrazione, spesso saranno necessari l'utilizzo di driver dato che tale strategia va ad integrare passo passo i sottosistemi partendo dal layer che si trova più in basso nella scala gerarchica.

La definizione dei test case avverrà tramite il framework JUnit, mentre verrà usato Mockito per il mocking e l'automatizzazione del run dei test sarà gestita da Maven.

## 6.3 Testing di sistema

Con il testing di sistema verrà effettuato un controllo della correttezza dell'intero sistema. Questo test sarà effettuato utilizzando il framework Selenium IDE , che mette a disposizione strumenti per il controllo di sistemi web based.

## 6.4 Sospensione e ripresa

Poichè il test del sistema richiede una quantità di tempo mediamente bassa per essere eseguita, non abbiamo ritenuto necessario definire i vincoli di sospensione e ripresa del testing.

## Ispezione del codice

Per aumentare la qualità del codice ci si affiderà principalmente ai controlli del tool Checkstyle.

## 7. Strumenti per il testing (hardware/software)

Per quanto riguarda il testing di sistema, non è prevista la presenza di hardware aggiuntivo per l'esecuzione dello stesso. E' sufficiente avere un computer su cui è installato.



## 8. Test cases

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

In questo paragrafo andremo ad illustrare le specifiche dei casi di test che adotteremo per il test di sistema. Come detto in precedenza, verrà adottata una strategia Category Partition.

### 9.1 Gestione cicli di dottorato

#### TC\_ 1.1 Gestione Registrazione

##### 1.1.1 Login

**Parametro:** Email

**Formato:**

(?:[a-z0-9!#\$%&'\*/+=?^\_`{|}~-]+(?:\.[a-z0-9!#\$%&'\*/+=?^\_`{|}~-]+)\*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*")@(?:(?:[a-z0-9](?:[a-z0-9]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9]\*[a-z0-9])?|[(?:(?25[0-5]|2[0-4][0-9]|01)?[0-9][0-9]?\.){3}(?25[0-5]|2[0-4][0-9]|01)?[0-9][0-9]?|[a-z0-9]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\))

Categorie	Scelte
Match [ME]	<ol style="list-style-type: none"><li>1. Match con email = false[error]</li><li>2. Match con email = true [PROPERTY ME_OK]</li></ol>



**Parametro:** Password

**Formato:** [A-Za-z] [0-9]

Categorie	Scelte
-----------	--------





<b>Match [MP]</b>	<b>1.</b> Match con password = false[error] <b>2.</b> Match con password = true [PROPERTY MP_OK]
-------------------	---

Codice	Combinazione	Esito
TC_1.1.1_01	ME1	errore
TC_1.1.1_02	ME2	inserimento
TC_1.1.1_03	ME2.MP1	errore
TC_1.1.1_04	ME2.MP2	inserimento

### 1.1.2 Registrazione cliente

<b>Parametro:</b> Nome <b>Formato:</b> [A-Za-z]	
Categorie	Scelte
<b>lunghezza Nome [LN]</b>	<b>1:</b> lunghezza < 2[errore] <b>2:</b> lunghezza 2-25 [property <b>lunghezzaLNok</b> , lunghezza nome da 2 a 25] <b>3:</b> lunghezza>25[errore]
<b>Formato [FN]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLNok</b> ][property <b>formatoFNok</b> , rispetta il formato [A-Za-z]] <b>2:</b> non rispetta il formato [if <b>lunghezzaLNok</b> ][errore]

<b>Parametro:</b> Cognome <b>Formato:</b> [A-Za-z]	
Categorie	Scelte
<b>Lunghezza [LC]</b>	<b>1:</b> lunghezza <2 [errore]



Laurea Magistrale in informatica-Università di Salerno Corso di  
*Gestione dei Progetti Software- Prof. C.Gravino*

	<b>2:</b> lunghezza 2-30 [property <b>lunghezzaLCok</b> , lunghezza cognome da 2 a 30] <b>3:</b> lunghezza >30 [errore]
<b>Formato [FC]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLCok</b> ] [property <b>formatoFCok</b> , rispetta il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLCok</b> ] [errore]

<b>Parametro:</b> Username <b>Formato:</b> [A-Za-z]	
Categorie	Scelte
<b>Lunghezza [LUS]</b>	<b>1:</b> lunghezza <2 [errore] <b>2:</b> lunghezza 2-16 [property <b>lunghezzaLUSok</b> , lunghezza cognome da 2 a 16] <b>3:</b> lunghezza >16 [errore]
<b>Formato [FUS]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLUSok</b> ] [property <b>formatoFUSok</b> , rispetta il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLUSok</b> ] [errore]

<b>Parametro:</b> Password <b>Formato:</b> [A-Za-z] [0-9]	
Categorie	Scelte
<b>Lunghezza [LP]</b>	<b>1:</b> lunghezza <8 [errore] <b>2:</b> lunghezza 8-16 [property <b>lunghezzaLPok</b> , lunghezza password tra 8 e 16] <b>3:</b> lunghezza >16 [errore]
<b>Formato [FP]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLPok</b> ] [property <b>formatoFPok</b> , rispecchia il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLPok</b> ] [errore]

<b>Parametro:</b> Ripeti Password <b>Formato:</b> [A-Za-z] [0-9]	
Categorie	Scelte



Laurea Magistrale in informatica-Università di Salerno Corso di  
*Gestione dei Progetti Software*- Prof. C.Gravino

<b>Lunghezza [LRP]</b>	<b>1:</b> lunghezza <8 [errore] <b>2:</b> lunghezza 8-16 [property <b>lunghezzaLPok</b> , lunghezza password tra 8 e 16] <b>3:</b> lunghezza >16 [errore]
<b>Formato [FRP]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLPok</b> ] [property <b>formatoFPok</b> , rispecchia il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLPok</b> ] [errore]

<b>Parametro:</b> Via e Numero Civico <b>Formato:</b> ^[0-9A-zÀ-ù ']{2,30}\$	
<b>Categorie</b>	<b>Scelte</b>
<b>formato [FVN]</b>	<b>1:</b> Rispetta il formato = false [error] <b>2:</b> Rispetta il formato = true [PROPERTY FVN_OK]

Codice	Combinazione	Esito
TC_1.1.2_01	LN2.FN1	inserimento
TC_1.1.2_02	LC1.FC2	errore
TC_1.1.2_03	LP1.FP2	errore
TC_1.1.2_04	LN2.LC2.LUS2.LP2.FVN1	inserimento

### 1.1.3 Inserimento Dipendente

<b>Parametro:</b> Nome <b>Formato:</b> [A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>



Laurea Magistrale in informatica-Università di Salerno Corso di  
*Gestione dei Progetti Software*- Prof. C.Gravino

<b>lunghezza Nome [LN]</b>	<b>1:</b> lunghezza < 2[errore] <b>2:</b> lunghezza 2-25 [property <b>lunghezzaLNok</b> , lunghezza nome da 2 a 25] <b>3:</b> lunghezza>25[errore]
<b>Formato [FN]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLNok</b> ][property <b>formatoFNok</b> , rispetta il formato [A-Za-z]] <b>2:</b> non rispetta il formato [if <b>lunghezzaLNok</b> ][errore]

<b>Parametro:</b> Cognome <b>Formato:</b> [A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
<b>Lunghezza [LC]</b>	<b>1:</b> lunghezza <2 [errore] <b>2:</b> lunghezza 2-30 [property <b>lunghezzaLCok</b> , lunghezza cognome da 2 a 30] <b>3:</b> lunghezza >30 [errore]
<b>Formato [FC]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLCok</b> ] [property <b>formatoFCok</b> , rispetta il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLCok</b> ] [errore]

<b>Parametro:</b> Username <b>Formato:</b> [A-Za-z]	
<b>Categorie</b>	<b>Scelte</b>
<b>Lunghezza [LUS]</b>	<b>1:</b> lunghezza <2 [errore] <b>2:</b> lunghezza 2-16 [property <b>lunghezzaLUSok</b> , lunghezza cognome da 2 a 16] <b>3:</b> lunghezza >16 [errore]
<b>Formato [FUS]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLUSok</b> ] [property <b>formatoFUSok</b> , rispetta il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLUSok</b> ] [errore]

<b>Parametro:</b> Password
----------------------------



<b>Formato:</b> [A-Za-z] [0-9]	
Categorie	Scelte
<b>Lunghezza [LP]</b>	<b>1:</b> lunghezza <8 [errore] <b>2:</b> lunghezza 8-16 [property <b>lunghezzaLPok</b> , lunghezza password tra 8 e 16] <b>3:</b> lunghezza >16 [errore]
<b>Formato [FP]</b>	<b>1:</b> rispetta il formato [if <b>lunghezzaLPok</b> ] [property <b>formatoFPok</b> , rispecchia il formato [A-Za-z] ] <b>2:</b> non rispetta il formato [if <b>lunghezzaLPok</b> ] [errore]

Codice	Combinazione	Esito
TC_1.1.3_01	LN2.FN2	errore
TC_1.1.3_02	LUS2.FUS1	inserimento
TC_1.1.3_03	LUS2.FUS2	errore
TC_1.1.3_04	LP1.FP2	errore



## TC\_ 1.2 Gestione Dipendente

### 1.2.1 Richiedi rifornimento

<b>Parametro:</b> Data consegna <b>Formato:</b> [DD-MM-YYYY]	
Categorie	Scelte
Inserimento[ID]	1: Data inserita rispetta il formato ed è successiva alla data corrente [if FD_OK && Data>DataCorrente][property ID_OK] 2: Data inserita rispetta il formato ma è precedente alla data corrente [if FD_OK && Data<=DataCorrente][errore]
Formato [FD]	1: Rispetta il formato [property FD_OK] 2: Non rispetta il formato [error]

<b>Parametro:</b> Quantità <b>Formato:</b> [0-9]	
Categorie	Scelte
Formato [FQ]	1: Rispetta il formato [property FQ_OK] 2: Non rispetta il formato [error]

Codice	Combinazione	Esito
TC_1.2.1_01	ID1.FD1.FQ1	inserimento
TC_1.2.1_02	ID1.FD2	errore



## TC\_ 1.3 Gestione Ricambi

### 1.3.1 Aggiorna ricambio

<b>Parametro:</b> Quantità <b>Formato:</b> [0-9]	
Categorie	Scelte
<b>Formato [FQ]</b>	1: Rispetta il formato [property FQ_OK] 2: Non rispetta il formato [error]

Codice	Combinazione	Esito
TC_1.3.1_01	FQ1	inserimento
TC_1.3.1_02	FQ2	errore

## 10. Pianificazione del testing e Assegnazione dei ruoli

Le attività di pianificazione del testing avverranno subito dopo la fase di design necessaria per la pianificazione.

La scrittura dei casi di test sarà effettuata in parallelo con la scrittura del codice.

L'esecuzione dei test avverrà sia durante che dopo l'implementazione del sistema. Una volta concluso lo sviluppo, tutti i test saranno rieseguiti per garantire il corretto funzionamento e produrre i report finali.