

# Università degli Studi di Salerno

## Corso di Ingegneria del Software

**Project: PC Zone**

**Test Plan**

**Document**

**Versione 2.0**

**PCZone**  
Pick Your Parts



Data: 02/01/2023

### Partecipanti

Nome	Matricola
Simone Scermino	0512110611
Roberto Andrei Miron	0512110581

### Revision History

Data	Versione	Descrizione	Autore
27/12/2022	1.0	Inizio stesura Test Plan	Simone Scermino, Roberto Andrei Miron
28/12/2022	1.1	Cosa testare, pass/fail criteria	Simone Scermino, Roberto Andrei Miron
30/12/2022	1.2	Approccio usato, materiali di testing, start/stop	Simone Scermino, Roberto Andrei Miron
02/01/2023	2.0	Rifinitura test plan	Simone Scermino, Roberto Andrei Miron

**1. Introduction**

**2. Relationship to other documents**

**3. System overview**

**4. Features to be tested/not to be tested**

**5. Pass/Fail criteria**

**6. Approach**

**7. Suspension and resumption**

**8. Testing materials (hardware/software requirements)**

**9. Test cases**

**10. Testing schedule**

# 1. Introduction

---

Questo documento si pone l'obiettivo di esplicitare la fase di testing del sistema PCZone, garantendo il deployment di un prodotto privo, quanto più possibile, di errori.

Saranno quindi identificate le relazioni con gli altri documenti di progetto, gli elementi e funzionalità da testare, specificando nella sezione 9 le specifiche di ciascun caso di test; eventuali errori riscontrati ci consentiranno di intervenire sull'implementazione, così da ridurre al minimo i fault prodotti.

## 2. Relationship to other documents

---

Per poter identificare correttamente i test case, ci si lega agli altri documenti prodotti durante le varie fasi di progettazione, così da poter avere informazioni sulla struttura generale del sistema.

In particolare:

- il RAD ci consente di avere una panoramica su requisiti funzionali e non funzionali, così da comprendere su cosa andare ad applicare i casi di test (e sulla priorità di essi)
- il SDD ci suggerisce come suddividere il test plan nei vari sottosistemi; nel nostro caso rispettando sia la struttura della architettura three-tier (Presentation, Application, Data), sia i sottosistemi da noi implementati.
- il ODD ci servirà come base d'appoggio per i testing, avendo una panoramica sulle interfacce delle classi implementate.

## 3. System overview

---

Nel SSD abbiamo descritto il tipo di approccio utilizzato per realizzare l'architettura del nostro sistema, ovvero three-tier, mantenendo la classica suddivisione in **Presentation, Application, Data** tier.

Nel presentation tier saranno presenti pagine che dovranno essere presentate all'utente tramite il browser da lui usato, e saranno realizzate con tecnologie quali HTML, CSS, JS.

Nell'applicazione tier avremo il controllo della logica applicativa, presente lato Server ed implementata tramite Java.

Nel data tier avremo l'uso di un Database relazionale SQL, gestito tramite il DBMS MySQL, al quale verrà effettuato l'accesso tramite apposito driver offerto da JDBC.

I sottosistemi identificati in quanto a funzionalità sono **GestioneUtente**, **GestioneSito**, **GestioneGestore**, **InterfacciaStorage**, **Database**

## 4. Features to be tested/not to be tested

---

Qui di seguito vengono riportate tutte le funzionalità le quali si è ritenuto opportuno sottoporre a test; si sono escluse tutte le funzionalità di visualizzazione e quelle con priorità bassa, prediligendo quindi quelle con input.

Funzionalità testate:

### **GestioneUtente**

- registraUtente()
- login()
- loginGestore()
- modificaInformazioni()

### **GestioneSito**

- recensisciComponente()
- assemblaPC()
- segnalaBug()

### **GestioneGestore**

- inserisciComponente()
- modificaComponente()
- inserisciGuida()
- modificaGuida()
- registrazioneGestore()

Funzionalità non testate:

### **GestioneUtente**

- logout()
- visualizzaProfilo()
- ripristinaPassword()

### **GestioneSito**

- visualizzaGuida()
- aggiungiFavoriti()
- visualizzaComponente()
- rimuoviFavoriti()
- visualizzaCatalogo()
- visualizzaElencoGuide()
- visualizzaElencoFavoriti()
- visualizzaHomepage()

### **GestioneGestore**

- rimuoviRecensione()

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

---

I criteri di successo e fallimento si basano sull'identificazione di faults all'interno del sistema, così da poter intervenire per la sua risoluzione prima del rilascio del sistema.

Basandoci quindi su un oracolo, ovvero il risultato atteso dell'esecuzione (che a sua volta si basa sull'analisi dei requisiti presenti nel RAD), diremo che il test avrà successo se il risultato ottenuto sarà differente da quello atteso; in caso contrario avremo un fallimento del test.

Da notare che questa definizione definisce tali successi e fallimenti basandosi sul concetto che il fulcro del testing è proprio l'identificazione (e successiva correzione) delle faults presenti.

Ad ogni risoluzione delle fault si effettueranno regression testing sul sistema, così da essere certi di non aver introdotto malfunzionamenti.

## **6. Approach**

---

L'attività di testing sarà effettuata mediante una combinazione di metodologie di testing di unità, testing di integrazione e testing di sistema. Nello specifico, durante il testing di sistema verrà esaminato individualmente ogni unità del sistema. Nel testing di integrazione, invece, verranno testate le interfacce delle unità testate nella fase precedente. Infine, il testing di sistema permetterà di verificare il funzionamento globale del sistema.

### **6.1 Testing di unità**

Durante questa fase di testing, verranno esaminate le condizioni di malfunzionamento del sistema isolando le singole componenti attraverso la tecnica Black-Box. Questa tecnica si concentra sul comportamento in input/output senza andare a vedere i meccanismi interni.

Gli input verranno suddivisi in classi di equivalenza e per ogni classe verrà utilizzato un solo caso di test. Questa tecnica viene chiamata Category Partition, che consente di scomporre gli input in categorie e per poi partizionare queste ultime in classi di equivalenza. Alla fine della fase di testing verranno create delle istanze di casi di test specificando i valori dei dati per ciascuna scelta e i loro risultati. Gli errori riscontrati durante le varie fasi di testing, verranno segnalati agli sviluppatori per consentire loro di apportare le modifiche necessarie e verranno descritti nel test incident report.

La fase di test di unità verrà condotta utilizzando JUnit e Mockito.

### **6.2 Testing di integrazione**

Durante questa fase, verrà utilizzata la metodologia nota come "Bottom-Up".

Il testing di integrazione bottom-up consiste nell'analizzare e verificare il funzionamento dei singoli componenti del sistema, partendo da quelli più semplici e piccoli. Questi componenti vengono successivamente uniti per formare dei componenti più complessi, che a loro volta verranno sottoposti a test. Questo processo continua fino a quando non si raggiunge il livello più alto del sistema, che verrà testato per accertare che tutti i componenti funzionino correttamente insieme. Questa tecnica è efficace nell'identificare eventuali

problemi di integrazione tra i componenti e garantire il corretto funzionamento del sistema come unità complessiva.

La fase di test di integrazione verrà condotta utilizzando JUnit e Mockito.

### **6.3 Testing di sistema**

Durante questa fase, si procederà con la verifica del funzionamento del sistema globale, al fine di accertare la conformità ai requisiti di priorità elevata richiesti. In particolare, verranno effettuati test sui servizi più utilizzati dall'utente e che presentano maggiori probabilità di generare malfunzionamenti. La fase di testing sistematico verrà condotta utilizzando il tool di automazione Selenium.

## **7. Suspension and resumption**

---

La sospensione dell'esecuzione dei casi di testing verrà effettuata quando malfunzionamenti riscontrati comporteranno modifiche a documenti precedenti o ad astrazioni di alto livello; nel Test Execution Report verranno riportate tale sospensioni sul test che le ha sollevate.

La ripresa della fase di testing comprende il regression testing sul sistema, così da essere certi di non aver introdotto malfunzionamenti.

## **8. Testing materials (hardware/software requirements)**

---

Gli strumenti che sono necessari per effettuare le attività di testing sono:

- DBMS MySQL per il database;
- WebServer Tomcat 9;
- Client-web usato per le operazioni da richiedere al sistema.

## **9. Test cases**

---

Vengono riportati di seguito i vari test case, suddivisi in base alle gestioni presenti. Le varie categorie di possibili input ci permettono di identificare i vari Test Frame da applicare.



## GestioneUtente

### RegistrazioneUtente (ID: RU)

<b>Parametro:</b> Nickname <b>Formato:</b> /[a-zA-Z0-9]{3,20}/	
Lunghezza LU	1. lunghezza < 3 OR lunghezza > 20 [LUnotok] 2. lunghezza >= 3 AND lunghezza <= 20 [LUok]
Formato FU	1. Non rispetta il formato [if LUok] [FUnotok] 2. Rispetta il formato [if LUok] [FUok]
Esiste EU	1. Esiste nel DB [if LUok AND FUok] [EUok] 2. Non esiste nel DB [if LUok AND FUok] [EUnook]

<b>Parametro:</b> Email <b>Formato:</b> /[a-zA-Z][a-zA-Z0-9.]*@([a-zA-Z]+).[a-zA-Z]+/	
Lunghezza LE	1. lunghezza < 1 [LEnotok] 2. lunghezza >= 1 [LEok]
Formato FE	3. Non rispetta il formato [if LEok] [FEnotok] 4. Rispetta il formato [if LEok] [FEok]
Esiste EE	1. Esiste nel DB [if LEok AND FEok] [EEok] 2. Non esiste nel DB [if LEok AND FEok] [EEnook]

<b>Parametro:</b> Password <b>Formato:</b> (?=\d)(?=[a-z])(?=.*[A-Z]).{8,}	
Lunghezza LP	1. lunghezza < 8 [LPnotok] 2. lunghezza >= 8 [LPok]
Formato FP	1. Non rispetta il formato [if LPok] [FPnotok] 2. Rispetta il formato [if LPok] [FPok]

<b>Parametro:</b> ConfermaPassword	
Match MP	1. Match con password corretto [MPok]

	2. Match con password errato [MPnotok]
--	--

Test case ID	Test frame	Esito
RU_TC1	LU1	Errore
RU_TC2	LU2, FU1	Errore
RU_TC3	LU2, FU2, EU1	Errore
RU_TC4	LU2, FU2, EU2, LE1	Errore
RU_TC5	LU2, FU2, EU2, LE2, FE1	Errore
RU_TC6	LU2, FU2, EU2, LE2, FE2, EE1	Errore
RU_TC7	LU2, FU2, EU2, LE2, FE2, EE2, LP1	Errore
RU_TC8	LU2, FU2, EU2, LE2, FE2, EE2, LP2, FP1	Errore
RU_TC9	LU2, FU2, EU2, LE2, FE2, EE2, LP2, FP2, MP1	Errore
RU_TC10	LU2, FU2, EU2, LE2, FE2, EE2, LP2, FP2, MP2	Ok

## Login id : LU

Parametro: Email	
Esiste EE	1. Non esiste nel DB [EEnotok] 2. Esiste nel DB [EEok]

Parametro: Password	
Matching MC	1. Non corrisponde con la password dell'email nel DB. [if EEok] [MCnotok] 2. Corrisponde con la password dell'email nel DB. [if EEok] [MCok]

Test case ID	Test frame	Esito
--------------	------------	-------

LU_TC1	EE1	Errore
LU_TC2	EE2,MC1	Errore
LU_TC3	EE2,MC2	Ok

**Login Gestore      id : LG**

<b>Parametro:</b> Email	
Esiste EE	<ol style="list-style-type: none"> <li>1. Non esiste nel DB [EEnotok]</li> <li>2. Esiste nel DB ma non corrisponde ad un account gestore [EEnotGest]</li> <li>3. Esiste nel DB e corrisponde ad un account gestore [EEok]</li> </ol>

<b>Parametro:</b> Password	
Matching MC	<ol style="list-style-type: none"> <li>1. Non corrisponde con la password dell'email nel DB. [ifEEok] [MCnotok]</li> <li>2. Corrisponde con la password dell'email nel DB. [ifEEok] [MCok]</li> </ol>

<b>Test case ID</b>	<b>Test frame</b>	<b>Esito</b>
LG_TC1	EE1	Errore
LG_TC2	EE2	Errore
LG_TC3	EE3, MC1	Errore
LG_TC4	EE3, MC2	Ok

**ModificaInformazioni ID: MI**

<b>Parametro:</b> Email <b>Formato:</b> /[a-zA-Z][a-zA-Z0-9.]*@([a-zA-Z]+).[a-zA-Z]+/	
Lunghezza LE	1. lunghezza < 1 [LEnotok] 2. lunghezza >= 1 [LEok]
Formato FE	1. Non rispetta il formato [if LEok] [FENotok] 2. Rispetta il formato [if LEok] [FEok]
Esiste EE	1. Esiste nel DB [if LEok AND FEok] [EEok] 2. Non esiste nel DB [if LEok AND FEok] [EEnook]

<b>Parametro:</b> Password <b>Formato:</b> (?=.\d)(?=[a-z])(?=.*[A-Z]).{8,}	
Lunghezza LP	1. lunghezza < 8 [LPnotok] 2. lunghezza >= 8 [LPok]
Formato FP	1. Non rispetta il formato [if LPok] [FPnotok] 2. Rispetta il formato [if LPok] [FPok]

<b>Parametro:</b> Descrizione <b>Formato:</b> /^.{0,1500}\$/	
Lunghezza LD	1. lunghezza > 1500 [LDnotok] 2. lunghezza <= 1500 [LDok]

<b>Parametro:</b> Immagine	
Lunghezza LI	1. lunghezza < 1 [LIempty] 2. lunghezza >= 1 [LIok]

Test case ID	Test frame	Esito
MI_TC1	LE1	Errore
MI_TC2	LE2, FE1	Errore

MI_TC3	LE2, FE2, EE1	Errore
MI_TC4	LE2, FE2, EE2, LP1	Errore
MI_TC6	LE2, FE2, EE2, LP2, FP1	Errore
MI_TC7	LE2, FE2, EE2, LP2, FP2, LD1	Errore
MI_TC8	LE2, FE2, EE2, LP2, FP2, LD2, LI1	ok
MI_TC9	LE2, FE2, EE2, LP2, FP2, LD2, LI2	ok

## GestioneSito

### recensisciComponente (id: RC)

<b>Parametro:</b> Titolo <b>Formato:</b> /^.{5,50}\$/	
Lunghezza LT	1. lunghezza < 5 OR lunghezza > 50 [LTnotok] 2. lunghezza >= 5 OR lunghezza <= 50 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> /^.{10,250}\$/	
Lunghezza LD	1. lunghezza < 10 OR lunghezza > 250 [LDnotok] 2. lunghezza >= 10 OR lunghezza <= 250 [LDok]

Test case ID	Test frame	Esito
RC_TC1	LT1	Errore
RC_TC2	LT2, LD1	Errore
RC_TC3	LT2, LD2	Ok

## SegnalaBug (id: SB)

<b>Parametro:</b> Titolo <b>Formato:</b> /^.{3,50}\$/	
Lunghezza LT	1. lunghezza < 3 OR lunghezza > 50 [LTnotok] 2. lunghezza >= 3 OR lunghezza <= 50 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> /^.{10,1000}\$/	
Lunghezza LD	1. lunghezza < 10 OR lunghezza > 1000 [LDnotok] 2. lunghezza >= 10 OR lunghezza <= 1000 [LDok]

Test case ID	Test frame	Esito
SB_TC1	LT1	Errore
SB_TC2	LT2, LD1	Errore
SB_TC3	LT2, LD2	Ok

## GestioneGestore

### InserisciComponente ID : IC

<b>Parametro:</b> Titolo <b>Formato:</b> /^.{10,70}\$/	
Lunghezza LT	1. lunghezza < 10 OR lunghezza > 70 [LTnotok] 2. lunghezza >= 10 OR lunghezza <= 70 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> /^.{10,750}\$/	
Lunghezza LD	1. lunghezza < 10 OR lunghezza > 750 [LDnotok] 2. lunghezza >= 10 OR lunghezza <= 750 [LDok]

<b>Parametro:</b> Immagine	
Lunghezza LI	1. lunghezza < 1 [LInotok] 2. lunghezza >= 1 [LIok]

<b>Parametro:</b> Prezzo <b>Formato:</b> / <sup>^</sup> .{0,30}\$ /	
Formato FP	1. formato non rispettato [FPnotok] 2. formato rispettato [FPok]

<b>Parametro:</b> LinkAcquisto	
Lunghezza LLA	1. lunghezza < 1 [LLAnotok] 2. lunghezza >= 1 [LLAok]

<b>Parametro:</b> DataUscita	
Selezionato SDU	1. Data di uscita non selezionata [SDUnotok] 2. Data di uscita selezionata [SDUok]

Test case ID	Test frame	Esito
IC_TC1	LT1	Errore
IC_TC2	LT2, LD1	Errore
IC_TC3	LT2, LD2, LI1	Errore
IC_TC4	LT2, LD2, LI2, FP1	Errore
IC_TC5	LT2, LD2, LI1, FP2, LLA1	Errore
IC_TC6	LT2, LD2, LI1, FP2, LLA2, SDU1	Errore
IC_TC7	LT2, LD2, LI1, FP2, LLA2, SDU2	Ok

**ModificaComponente ID : MC**

<b>Parametro:</b> Titolo <b>Formato:</b> / <sup>^</sup> .{10,70}\$/ 	
Lunghezza LT	1. lunghezza < 10 OR lunghezza > 70 [LTnotok] 2. lunghezza >= 10 OR lunghezza <= 70 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> / <sup>^</sup> .{10,750}\$/ 	
Lunghezza LD	1. lunghezza < 10 OR lunghezza > 750 [LDnotok] 2. lunghezza >= 10 OR lunghezza <= 750 [LDok]

<b>Parametro:</b> Immagine 	
Lunghezza LI	1. lunghezza < 1 [LInotok] 2. lunghezza >= 1 [LIok]

<b>Parametro:</b> Prezzo <b>Formato:</b> [0-9]+,[0-9]{2}	
Formato FP	1. formato non rispettato [FPnotok] 2. formato rispettato [FPok]

<b>Parametro:</b> LinkAcquisto	
Lunghezza LLA	1. lunghezza < 1 [LLAnotok] 2. lunghezza >= 1 [LLAok]

<b>Parametro:</b> DataUscita	
Selezionato SDU	1. Data di uscita non selezionata [SDUnotok] 2. Data di uscita selezionata [SDUok]

Test case ID	Test frame	Esito
IC_TC1	LT1	Errore



IC_TC2	LT2, LD1	Errore
IC_TC3	LT2, LD2, LI1	Errore
IC_TC4	LT2, LD2, LI2, FP1	Errore
IC_TC5	LT2, LD2, LI1, FP2, LLA1	Errore
IC_TC6	LT2, LD2, LI1, FP2, LLA2, SDU1	Errore
IC_TC7	LT2, LD2, LI1, FP2, LLA2, SDU2	Ok

**inserisciGuida    ID : IG**

<b>Parametro:</b> Titolo <b>Formato:</b> /^.{10,50}\$/	
Lunghezza LT	1. lunghezza < 10 OR lunghezza > 50 [LTnotok] 2. lunghezza >= 10 OR lunghezza <= 50 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> /^.{10,1500}\$/	
Lunghezza LD	1. lunghezza < 10 OR lunghezza > 1500 [LDnotok] 2. lunghezza >= 10 OR lunghezza <= 1500 [LDok]

Test case ID	Test frame	Esito
IG_TC1	LT1	Errore
IG_TC2	LT2, LD1	Errore
IG_TC3	LT2, LD2	Ok

**modificaGuida ID : MG**

<b>Parametro:</b> Titolo <b>Formato:</b> / <sup>^</sup> .{10,50}\$/ 	
Lunghezza LT	3. lunghezza < 10 OR lunghezza > 50 [LTnotok] 4. lunghezza >= 10 OR lunghezza <= 50 [LTok]

<b>Parametro:</b> Descrizione <b>Formato:</b> / <sup>^</sup> .{10,1500}\$/ 	
Lunghezza LD	3. lunghezza < 10 OR lunghezza > 1500 [LDnotok] 4. lunghezza >= 10 OR lunghezza <= 1500 [LDok]

Test case ID	Test frame	Esito
MG_TC1	LT1	Errore
MG_TC2	LT2, LD1	Errore
MG_TC3	LT2, LD2	Ok

**Registrazione Gestore id: RG**

<b>Parametro:</b> Nickname <b>Formato:</b> /[a-zA-Z0-9]{3,20}/ 	
Lunghezza LU	1. lunghezza < 3 OR lunghezza > 20 [LUnotok] 2. lunghezza >= 3 AND lunghezza <= 20 [LUok]
Formato FU	1. Non rispetta il formato [if LUok] [FUnotok] 2. Rispetta il formato [if LUok] [FUok]
Esiste EU	1. Esiste nel DB [if LUok AND FUok] [EUok] 2. Non esiste nel DB [if LUok AND FUok] [EUnook]

<b>Parametro:</b> Email <b>Formato:</b> /[a-zA-Z][a-zA-Z0-9.]*@([a-zA-Z]+).[a-zA-Z]+/ 	
Lunghezza LE	1. lunghezza < 1 [LEnotok]

	2. lunghezza $\geq 1$ [LEok]
Formato FE	1. Non rispetta il formato [if LEok] [FEnotok] 2. Rispetta il formato [if LEok] [FEok]
Esiste EE	1. Esiste nel DB [if LEok AND FEok] [EEok] 2. Non esiste nel DB [if LEok AND FEok] [EEnook]

<b>Parametro:</b> Password <b>Formato:</b> (?=\d)(?=[a-z])(?=[A-Z]).{8,}	
Lunghezza LP	1. lunghezza $< 8$ [LPnotok] 2. lunghezza $\geq 8$ [LPok]
Formato FP	1. Non rispetta il formato [if LPok] [FPnotok] 2. Rispetta il formato [if LPok] [FPok]

Test case ID	Test frame	Esito
RG_TC1	LU1	Errore
RG_TC2	LU2, FU1	Errore
RG_TC3	LU2, FU2, EU1	Errore
RG_TC4	LU2, FU2, EU2, LE1	Errore
RG_TC5	LU2, FU2, EU2, LE2, FE1	Errore
RG_TC6	LU2, FU2, EU2, LE2, FE2, EE1	Errore
RG_TC7	LU2, FU2, EU2, LE2, FE2, EE2, LP1	Errore
RG_TC8	LU2, FU2, EU2, LE2, FE2, EE2, LP2, FP1	Errore
RG_TC9	LU2, FU2, EU2, LE2, FE2, EE2, LP2, FP2	Ok

## **10. Testing schedule**

---

La scrittura dei casi di testing avverrà contemporaneamente allo sviluppo del codice.

Il testing di ogni unità verrà avviato in seguito all'implementazione di essa, mentre per quanto riguarda il testing di integrazione e sistema si dovrà attendere la conclusione dei precedenti test di unità.