



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino



# TP

## Test Plan

# SmartGym

Riferimento	thISTeam_TP
Versione	2.0
Data	17/02/2023
Destinatario	Docente di Ingegneria del Software 2022/23 (Carminè Gravino)
Presentato da	thISTeam
Approvato da	



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## Revision History

Data	Versione	Descrizione	Autori
07/12/2022	0.1	Creazione del documento	thlSTeam
12/12/2022	0.2	Stesura iniziale dei vari capitoli	thlSTeam
15/12/2022	0.3	Aggiunta della stesura mancante del capitolo 6	thlSTeam
17/12/2022	1.0	Revisione	thlSTeam
13/02/2023	1.1	Aggiunti dettagli relativi alle relazioni con i nuovi documenti	thlSTeam
17/02/2023	2.0	Revisione	thlSTeam



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## Team Members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Daniele Fabiano	Team Member	DF	d.fabiano3@studenti.unisa.it
Mariantonietta Maselli	Team Member	MM	m.maselli3@studenti.unisa.it
Michele Spinelli	Team Member	MS	m.spinelli26@studenti.unisa.it



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## Indice

<b>1. Introduzione</b>	<b>5</b>
<b>2. Relazione con altri documenti</b>	<b>5</b>
<b>4. Features da testare/da non testare</b>	<b>6</b>
<b>5. Pass/Fail criteria</b>	<b>7</b>
<b>6. Approccio</b>	<b>7</b>
<b>7. Sospensione e ripristino</b>	<b>8</b>
<b>8. Materiale di testing</b>	<b>8</b>
<b>9. Test Cases</b>	<b>10</b>
<b>10. Testing schedule</b>	<b>13</b>



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## 1. Introduzione

---

SmartGym propone di semplificare la distribuzione e la fruizione della scheda esercizi degli atleti iscritti alla palestra, al fine di ridurre l'uso del cartaceo creando uno strumento di digitalizzazione dei processi che portano alla realizzazione di una scheda esercizi.

Il documento di Test Plan si pone come obiettivo quello di descrivere e analizzare le attività di Testing per la piattaforma SmartGym, al fine di garantire che tutti gli aspetti del sistema funzionino correttamente.

All'interno del documento sono riportate le strategie di testing adottate, quali funzionalità saranno testate e gli strumenti scelti per la rilevazione degli errori, con lo scopo di presentare al cliente finale un'applicazione priva di malfunzionamenti. Sono state pianificate attività di testing per le seguenti gestioni:

- Gestione Informazioni Utenti
- Gestione Scheda Esercizi
- Gestione Assistenza Allenamento

## 2. Relazione con altri documenti

---

### **Relazioni con il Requirements Analysis Document (RAD)**

I test case pianificati nel Test Plan sono elaborati in relazione ai requisiti funzionali e non funzionali presentati nel RAD.

### **Relazioni con il System Design Document (SDD)**

I test case pianificati nel Test Plan devono rispettare la suddivisione in sottosistemi presentata nell'SDD.



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## Relazioni con Object Design Document (ODD)

I test di unità che saranno effettuati fanno riferimento alle operazioni delle classi definite nel documento di Object Design che saranno poi tradotte nell'implementazione come i metodi su cui effettuare il testing

## 3. Panoramica del sistema

---

Il sistema proposto basa la sua architettura sul three-tier.

Verranno usate le tecnologie standard dello sviluppo Android per la parte di front-end (XML).

Per la logica applicativa e la parte di back-end si farà uso del linguaggio Java e delle API fornite dal servizio di database FireStore.

## 4. Features da testare/da non testare

---

Di seguito la lista delle features di cui si effettuerà il testing per le varie gestioni:

- Gestione Informazioni Utente
  - Login
  - Inserimento Caratteristiche Atleta
- Gestione Scheda Esercizi
  - Creazione Scheda Esercizi
- Gestione Assistenza Allenamento

Il numero di funzionalità testate sono in rispetto ai vincoli prestabiliti dal progetto. Inoltre non verranno testate funzionalità della Gestione Assistenza Allenamento. Le motivazioni dietro questa scelta riguardano costi di tempo per la consegna che non permetteranno la realizzazione di questo gruppo di funzionalità e del relativo sottosistema. Il testing di queste funzionalità sarà oggetto di una futura nuova major release del sistema.



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## 5. Pass/Fail criteria

---

Le attività di testing sono mirate ad identificare la presenza di faults (errori) all'interno del sistema, per effettuare un successivo intervento di eliminazione.

L'esito di un test case è valutato mediante un oracolo, inteso come il risultato atteso della sua esecuzione, basandosi sui requisiti.

Un test ha successo (*pass*) se, dato un input al sistema, l'output ottenuto è diverso dall'output atteso dall'oracolo.

Un test fallisce (*fail*) se, dato un input al sistema, l'output ottenuto è uguale all'output atteso dall'oracolo.

Tutto il testing sarà considerato valido una volta completate le attività inerenti i vincoli di consegna.

## 6. Approccio

---

### Testing di unità

Si concentra su una componente del sistema isolandola ed effettuandone il testing. Nel nostro caso si effettuerà testing di unità sui metodi delle classi che devono essere testate.

### Testing di sistema

Si concentra sull'intero sistema partendo dai suoi requisiti funzionali/non funzionali.

### Functional Testing

Cerca le differenze tra i requisiti funzionali individuati e il sistema realizzato. Si effettua l'individuazione di test case che fanno uso di una tecnica black box.



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

Il black box testing si occupa di individuare gli input necessari alla funzionalità e ottenere gli output attesi, senza preoccuparsi degli aspetti interni, del comportamento e della struttura della componente testata.

Un metodo per effettuare questa tipologia di testing è il category partition.

Il category partition suddivide il dominio di input definendo una classe individuata dal relativo parametro da testare.

Ogni parametro ricade in una specifica categoria e a seconda dei suoi vincoli si stabilisce quando il test ha successo o fallisce.

Questa metodologia viene messa in pratica applicandola sulla specifica di ogni test case individuato.

## 7. Sospensione e ripristino

---

### **Criteri di sospensione**

Il testing non verrà sospeso fino alla sua terminazione, anche in caso di rilevazione di una failure. Il testing potrà essere momentaneamente sospeso nel caso venga restituito, al momento dell'esecuzione, un errore nella definizione di uno dei test stessi.

### **Criteri di ripristino**

Il testing verrà ripreso dopo aver risolto i fault individuati.

## 8. Materiale di testing

---

Per effettuare le attività di testing è necessario un PC ed uno smartphone sul quale installare e testare la app. In alternativa allo smartphone è possibile utilizzare l'emulatore integrato nell'ambiente di sviluppo Android. Per il testing di unità sarà utilizzato il framework Junit, mentre per quello di sistema sarà utilizzato il framework [Espresso](#).





Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## 9. Test Cases

### 9.1 Gestione Informazioni Utenti

#### 9.1.1 Login

Parametro: Email	
Nome Categoria	Scelte per la categoria
Match [ME]	1. Match con email = false [error] 2. Match con email = true [PROPERTY_ME_OK]
Parametro: Password	
Nome Categoria	Scelte per la categoria
Match [MP]	1. Match con password = false [error] 2. Match con password = true [PROPERTY_MP_OK]

Test Case ID	Test Frame	Esito
TC_1.1	ME1	Errato: email errata
TC_1.2	ME2, MP1	Errato: Errato: password non corretta
TC_1.3	ME2, MP2	Corretto



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

### 9.1.2 Inserimento Caratteristiche Atleti

Parametro: Peso	
Nome Categoria	Scelte per la categoria
Correttezza [CP]	1. $\text{Peso} < 40 \text{ OR } \text{Peso} > 300$ [errore] 2. $\text{Peso} \geq 40 \text{ AND } \text{Peso} \leq 300$ [PROPERTY_CP_OK]
Parametro: Altezza	
Nome Categoria	Scelte per la categoria
Correttezza [CA]	1. $\text{Altezza} < 130 \text{ OR } \text{Altezza} > 230 = \text{false}$ [errore] 2. $\text{Altezza} \geq 130 \text{ AND } \text{Altezza} \leq 230 = \text{true}$ [PROPERTY_CA_OK]
Parametro: Numero Allenamenti	
Nome Categoria	Scelte per la categoria
Correttezza [CN]	1. $\text{NumAllenamenti} < 2 \text{ OR } \text{NumAllenamenti} > 7 = \text{false}$ [errore] 2. $\text{NumAllenamenti} \geq 2 \text{ AND } \text{NumAllenamenti} \leq 7 = \text{true}$ [PROPERTY_CN_OK]

Test Case ID	Test Frame	Esito
TC_2.1	CP1	Errato: peso non valido
TC_2.2	CP2	Errato: peso non valido
TC_2.3	CP3, CA1	Errato: altezza non valida
TC_2.4	CP3, CA2	Errato: altezza non valida
TC_2.5	CP3, CA3, CN1	Errato: numero allenamenti non valido
TC_2.6	CP3, CA3, CN2	Errato: numero allenamenti non valido
TC_2.7	CP3, CA3, CN3	Corretto



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## 9.2 Gestione Scheda Esercizi

### 9.2.1 Creazione Scheda Esercizi

Parametro: Ripetizioni Esercizio	
Nome Categoria	Scelte per la categoria
Correttezza [CR]	1. $(\text{Ripetizioni} < 3 \text{ OR } \text{Ripetizioni} > 30) = \text{false}$ [error] 2. $(\text{Ripetizioni} \geq 3 \text{ AND } \text{Ripetizioni} \leq 30) = \text{true}$ [PROPERTY_CR_OK]
Parametro: Durata Esercizio (in secondi)	
Nome Categoria	Scelte per la categoria
Correttezza [CD]	1. $(\text{Durata} < 20 \text{ OR } \text{Durata} > 60) = \text{false}$ [error] 2. $(\text{Durata} \geq 20 \text{ AND } \text{Durata} \leq 60) = \text{true}$ [PROPERTY_CD_OK]
Parametro: Numero esercizi	
Nome Categoria	Scelte per la categoria
Correttezza [CN]	1. $(\text{Numero Esercizi} < 3 \text{ OR } \text{Numero Esercizi} > 10) = \text{false}$ [error] 2. $(\text{Numero Esercizi} \geq 3 \text{ AND } \text{Numero Esercizi} \leq 10) = \text{true}$ [PROPERTY_CN_OK]

Test Case ID	Test Frame	Esito
TC_3.1	CR1	Errato: Ripetizioni non valide
TC_3.2	CR2	Errato: Ripetizioni non valide
TC_3.3	CR3, CD1	Errato: Durata non valida
TC_3.4	CR3, CD2	Errato: Durata non valida
TC_3.5	CR3, CD3, CN1	Errato: Numero esercizi non valido
TC_3.6	CR3, CD3, CN2	Errato: Numero esercizi non valido
TC_3.7	CR3, CD3, CN3	Corretto



Laurea Triennale in Informatica - Università di Salerno  
Corso di Ingegneria del Software - Prof. C.Gravino

## 10. Testing schedule

---

La stesura dei casi di test sarà effettuata prima della realizzazione dello sviluppo dell'applicazione.

L'esecuzione dei casi di test potrà avvenire durante l'implementazione una volta terminata la funzionalità da testare oppure alla fine della completa implementazione.