



# TP

## Test Plan

### Stay Up

Riferimento	NC8_TP
Versione	1.0
Data	14/01/2023
Destinatario	Prof. Carmine Gravino
Presentato da	Abbate Andrea, Annunziata Nicola, Della Corte Gaetano, Sulipano Raffaele
Approvato da	



### Revision History

Data	Versione	Descrizione	Autori
04/12/2023	0.1	Prima stesura	RS
03/01/2024	0.2	Modifica Test Case Gestione Account	RS
05/01/2024	0.3	Inserimento TC_2.1	RS
14/01/2024	1.0	Revisione del document	Tutto il team

### Team Members

Nome	Ruolo progetto	Acronimo	Informazione di contatto
Giammaria Giordano	Project Manager	GG	giagiordano@unisa.it
Andrea Abbate	Team Member	AA	a.abbate20@studenti.unisa.it
Nicola Annunziata	Team Member	NA	n.annunziata12@studenti.unisa.it
Gaetano Della Corte	Team Member	GD	g.dellacorte13@studenti.unisa.it
Raffaele Sulipano	Team Member	RS	r.sulipano@studenti.unisa.it



## Sommario

1. Introduzione.....	4
2. Riferimenti.....	4
3. Panoramica del sistema.....	5
4. Feature da testare.....	5
5. Feature da non testare.....	5
6. Pass/Fail Criteria .....	6
7. Approccio .....	6
8. Sospensione e Ripristino .....	7
9. Materiale di Testing.....	8
10. Test Cases.....	9
11. Scheduling delle attività di testing.....	22



## 1. Introduzione

---

StayUp si propone di semplificare il lavoro del Personal Trainer e la relazione tra Personal Trainer ed Utente, agevolando la creazione e l'invio di schede personalizzate con la possibilità di consultare lo storico delle schede di allenamento, la consultazione dei propri parametri per tenere traccia del proprio progresso.

Il seguente documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di testing per il sistema StayUp, al fine di garantire il corretto funzionamento del sistema. All'interno del documento sono indicate le strategie di testing adottate, le funzionalità da testare e gli strumenti scelti che saranno utilizzati per la rilevazione degli errori, con lo scopo di presentare sul mercato un sistema utilizzabile e privo di malfunzionamenti. Le attività di testing sono state pianificate per le seguenti gestioni:

- Gestione Utente
- Gestione Scheda di Allenamento
- Gestione Abbonamento

## 2. Riferimenti

---

Tale documento è connesso a tutti i documenti generati fino al rilascio del sistema. I casi di test si basano sulle funzionalità del sistema, identificate e riassunte nei seguenti documenti:

- **Relazioni con il Documento di Analisi dei Requisiti (RAD)**

I casi di test pianificati nel Piano di Test sono sviluppati in relazione ai requisiti funzionali e non funzionali presentati nel Documento di Analisi dei Requisiti (RAD).

- **Relazioni con il Documento di Progettazione del Sistema (SDD)**

I casi di test pianificati nel Piano di Test devono rispettare la suddivisione in sottosistemi presentata nel Documento di Progettazione del Sistema (SDD).

### 3. Panoramica del sistema

Il sistema proposto adotta un'architettura basata sul modello Client-Server. Nello specifico, per lo sviluppo, si fa uso della tecnologia JSP, la cui architettura è strutturata secondo il modello MVC (Model-View-Controller). Nel backend, troviamo l'implementazione tramite Servlet e classi Java, mentre per il frontend si opta per la creazione delle View utilizzando linguaggi come HTML5, CSS3, Java e JavaScript. In aggiunta, si sceglie di integrare il framework Bootstrap per velocizzare lo sviluppo delle View, garantendo tempi ridotti. Per la gestione dei dati, si adotta un database MySQL.

La comunicazione tra il backend e il frontend avviene attraverso richieste HTTP, che vengono elaborate e gestite dalle Servlet all'interno del backend.

### 4. Feature da testare

Le features da testare per ogni gestione saranno le seguenti:

- Gestione Utente
  - Autenticazione **Utente**
  - Gestione Parametri Fisici
  - Gestione Account
    - Cambio Email
    - Cambio Password
    - Cambio Nome e Cognome
    - Cambio Numero
    - Cambio Sesso
- Gestione Scheda di Allenamento
  - Realizzazione Scheda di Allenamento

### 5. Feature da non testare

Le funzionalità escluse dal testing riguardano i requisiti funzionali di bassa e media priorità o i requisiti per cui non è necessario creare dei casi di test perché non accettano input utente particolari (ad esempio le funzionalità di visualizzazione).

Di seguito sono elencate le funzionalità da non testare:

- Gestione Utente
  - Gestione Abbonamento
  - Accettazione Personal Trainer
- Gestione Scheda di Allenamento
  - Visualizzazione Scheda di Allenamento
  - Richiesta Scheda di Allenamento

- Gestione Abbonamenti
  - Visualizzazione Informazioni Cliente
  - Visualizzazione Informazioni Personal Trainer
  - Visualizzazione Abbonamento
  - Accettazione Abbonamento
  - Richiesta Abbonamento

## 6. Pass/Fail Criteria

Le attività di testing sono finalizzate a rilevare la presenza di difetti (errori) all'interno del sistema, al fine di eseguire interventi correttivi successivi. L'esito di ciascun caso di test è valutato attraverso un oracolo, rappresentato dal risultato atteso della sua esecuzione, basato sui requisiti.

Un test è considerato:

- riuscito (**passed**) quando, dato un input al sistema, l'output ottenuto è identico a quello previsto dall'oracolo.
- fallito (**failed**) se, dato un input al sistema, l'output ottenuto è diverso da quello atteso dall'oracolo.

Il testing sarà ritenuto valido se saranno rispettati i seguenti vincoli:

- Testare tutti i requisiti ad alta priorità elencati nel capitolo precedente.
- Effettuare test di regressione ogni volta che si introducono nuove caratteristiche al sistema o vengono modificate quelle presenti.
- Raggiungere un branch coverage non inferiore al 75%

## 7. Approccio

Il processo di testing del sistema si articola in tre fasi, eseguite in ordine inverso e progettate seguendo la sequenza seguente:

- Testing di Sistema
- Testing di Integrazione
- Testing di Unità

La progettazione dei casi di test di sistema e di integrazione avviene prima della fase di implementazione del sistema e viene successivamente affinata durante la loro esecuzione. Al contrario, la progettazione dei casi di test di unità avviene concomitantemente alla fase di implementazione.

L'esecuzione dei test avviene in parallelo alla stesura del codice, consentendo un immediato feedback sugli errori di implementazione. Ogni volta che viene introdotta una nuova funzionalità, vengono eseguiti i test correlati ad essa e, inoltre, i test di integrazione associati.

Ogni volta che viene apportata una modifica alle componenti, saranno eseguiti i test di unità e di integrazione delle componenti direttamente correlate a quelle modificate. In aggiunta, sarà condotto un testing di regressione mediante la ripetizione dei casi di test delle componenti apparentemente non influenzate dalla modifica, al fine di prevenire il manifestarsi di effetti collaterali indesiderati. Tale pratica



mira a mitigare il potenziale effetto a catena (Ripple Effect) che potrebbe derivare dalle modifiche apportate.

## **7.1. Testing di Sistema**

Per il testing di sistema, verrà impiegato il tool Selenium IDE. Questo strumento consente di registrare le azioni eseguite da un utente sul browser al fine di creare e eseguire i casi di test di sistema. Inoltre, il server sarà distribuito in ambiente localhost per condurre i test.

## **7.2. Testing di Integrazione**

L'approccio prescelto è il bottom-up, considerato il più adatto per i software basati sul paradigma Object Oriented. Per la definizione dei casi di test in Java, sarà impiegato il framework JUnit, mentre per la simulazione delle componenti verrà utilizzato Mockito. Inoltre, al fine di soddisfare i requisiti di premialità, sarà adottata l'integrazione continua mediante l'utilizzo delle GitHub Actions. L'esecuzione automatizzata dei test sarà gestita da Maven, mentre per la valutazione della copertura del codice sarà utilizzato il tool JaCoCo, il quale fornirà misurazioni precise e genererà report esaustivi sulla copertura del codice.

## **7.3. Testing di Unità**

Per il processo di testing di unità, si procederà alla verifica di ciascun metodo all'interno di ogni classe del sistema, con l'eccezione del testing unitario per le interfacce e le classi di dominio. Queste ultime, caratterizzate esclusivamente da metodi getter e setter generati automaticamente dall'IDE, non richiederanno un testing dedicato.

La definizione dei casi di test unitari seguirà un approccio funzionale (black-box). Per implementare questa metodologia, si farà uso del framework JUnit. Ogni classe sorgente sarà accompagnata da una classe di test corrispondente.

Qualora l'esecuzione dei test unitari non dovesse raggiungere il livello minimo di copertura del codice di ramo richiesto (75%), si procederà alla definizione di ulteriori casi di test con un approccio strutturale (white-box).

Come per il testing di integrazione, saranno utilizzati strumenti come Mockito, JaCoCo, Maven e GitHub Actions per raggiungere gli obiettivi di testing prefissati.

Da notare che l'esecuzione dei test unitari sarà pianificata in anticipo rispetto a quella dei test di integrazione, al fine di garantire una corretta sequenza di test nel processo di sviluppo del software.

# **8. Sospensione e Ripristino**

Nella presente sezione, saranno delineati i criteri per la sospensione del processo di test e le attività che richiederanno una ripetizione al momento del suo ripristino.

## **8.1. Criteri di Sospensione**

Il processo di testing sarà condotto senza interruzioni fino al suo completo espletamento, anche in presenza di eventuali anomalie rilevate. La sospensione temporanea del testing avverrà soltanto qualora si verifichi un errore nella definizione di uno dei test durante l'esecuzione.



## 8.2. Criteri di Ripristino

La ripresa del processo di testing avverrà successivamente alla correzione dei difetti individuati.

# 9. Materiale di Testing

---

Per condurre l'attività di testing, è richiesto un computer in possesso dell'ultima versione del sistema ottenuta attraverso la repository del progetto.



## 10. Test Cases

Per individuare i casi di test elencati di seguito, si è ricorsi alla metodologia del Category Partitioning e alla definizione delle Classi di Equivalenza.

### 10.1. Gestione Utente

#### 10.1.1. Autenticazione Utente

Parametro: Email utente Formato: <code>^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,})+\$</code>	
Lunghezza [LE]	<ol style="list-style-type: none"> <li>&lt;6 or &gt;40 [error]</li> <li>&gt;=6 and &lt;=40 [property LE_OK]</li> </ol>
Formato [FE]	<ol style="list-style-type: none"> <li>Non match [if LE_OK] [error]</li> <li>Match [if LE_OK] [property FE_OK]</li> </ol>

Parametro: Password utente Formato: <code>^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&amp;])[A-Za-z\d@\$!%*?&amp;]{8,24}\$</code>	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>&lt;8 or &gt;24 [error]</li> <li>&gt;=8 and &lt;=24 [property LE_OK]</li> </ol>
Formato [FP]	<ol style="list-style-type: none"> <li>Non match [if LE_OK] [error]</li> <li>Match [if LE_OK] [property FE_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.1_1	LE1, FE2, LP2, FP2	Errore: lunghezza email non valida
TC_1.1_2	LE2, FE1, LP2, FP2	Errore: formato email non valido
TC_1.1_3	LE2, FE2, LP1, FP2	Errore: lunghezza password non valida
TC_1.1_4	LE2, FE2, LP2, FP1	Errore: formato password non valido
TC_1.1_5	LE2, FE2, LP2, FP2	Corretto: tentativo di autenticazione utente

### 10.1.2. Gestione Parametri Fisici

Parametro: Peso	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LP_OK]</li> </ol>
Formato [FP]	<ol style="list-style-type: none"> <li>1. Non match [if LP_OK] [error]</li> <li>2. Match [if LP_OK] [property FP_OK]</li> </ol>

Parametro: Massa Magra	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LMM]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LMM_OK]</li> </ol>
Formato [FMM]	<ol style="list-style-type: none"> <li>1. Non match [if LMM_OK] [error]</li> <li>2. Match [if LMM_OK] [property FMM_OK]</li> </ol>

Parametro: Massa Grassa	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LMG]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LMG_OK]</li> </ol>
Formato [FMG]	<ol style="list-style-type: none"> <li>1. Non match [if LMG_OK] [error]</li> <li>2. Match [if LMG_OK] [property FMG_OK]</li> </ol>

Parametro: Misurazione Braccia	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LMB]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LMB_OK]</li> </ol>
Formato [FMB]	<ol style="list-style-type: none"> <li>1. Non match [if LMB_OK] [error]</li> <li>2. Match [if LMB_OK] [property FMB_OK]</li> </ol>

Parametro: Misurazione Gambe	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LMGA]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LMGA_OK]</li> </ol>
Formato [FMGA]	<ol style="list-style-type: none"> <li>1. Non match [if LMGA_OK] [error]</li> <li>2. Match [if LMGA_OK] [property FMGA_OK]</li> </ol>

Parametro: Misurazione Petto	
Formato: $^{(?=\d{2,3}(?:\.\d{0,2})?)}[0-9]*(?:\.\d*)?$$	
Lunghezza [LMP]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;3 [error]</li> <li>2. &gt;= 2 and &lt;=3 [property LMP_OK]</li> </ol>
Formato [FMP]	<ol style="list-style-type: none"> <li>1. Non match [if LMP_OK] [error]</li> <li>2. Match [if LMP_OK] [property FMP_OK]</li> </ol>

Parametro: Misurazione Fianchi	
Formato: $^(?=\d{2,3}(?:\.\d{0,2})?)[0-9]*(?:\.\d*)?$$	
Lunghezza [LMF]	<ol style="list-style-type: none"> <li>&lt;2 or &gt;3 [error]</li> <li><math>\geq 2</math> and <math>\leq 3</math> [property LMF_OK]</li> </ol>
Formato [FMF]	<ol style="list-style-type: none"> <li>Non match [if LMF_OK] [error]</li> <li>Match [if LMF_OK] [property FMF_OK]</li> </ol>

Parametro: Misurazione Spalle	
Formato: $^(?=\d{2,3}(?:\.\d{0,2})?)[0-9]*(?:\.\d*)?$$	
Lunghezza [LMS]	<ol style="list-style-type: none"> <li>&lt;2 or &gt;3 [error]</li> <li><math>\geq 2</math> and <math>\leq 3</math> [property LMS_OK]</li> </ol>
Formato [FMS]	<ol style="list-style-type: none"> <li>Non match [if LMS_OK] [error]</li> <li>Match [if LMS_OK] [property FMS_OK]</li> </ol>

Parametro: Anni di Allenamento	
Formato: $^[0-9]\{1,2\}$	
Lunghezza [LAA]	<ol style="list-style-type: none"> <li>&lt;1 or &gt;2 [error]</li> <li><math>\geq 1</math> and <math>\leq 2</math> [property LAA_OK]</li> </ol>
Formato [FAA]	<ol style="list-style-type: none"> <li>Non match [if LAA_OK] [error]</li> <li>Match [if LAA_OK] [property FAA_OK]</li> </ol>

Parametro: Password utente	
Formato: $^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]\{8,24\}$	
Lunghezza [LPS]	<ol style="list-style-type: none"> <li>&lt;8 or &gt;24 [error]</li> <li><math>\geq 8</math> and <math>\leq 24</math> [property LE_OK]</li> </ol>
Formato [FPS]	<ol style="list-style-type: none"> <li>Non match [if LE_OK] [error]</li> <li>Match [if LE_OK] [property FE_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.2_1	LP1, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB1, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza peso non valida
TC_1.2_2	LP2, FP1, LMM2, FMM2, LMG2, FMG2, LMB2, FMB1, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato peso non valido
TC_1.2_3	LP2, FP2, LMM1, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA1, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2,	Errore: lunghezza massa magra non valida

	FMS2, LAA2, FAA2, LPS2, FPS2	
TC_1.2_4	LP2, FP2, LMM2, FMM1, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA1, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato massa magra non valido
TC_1.2_5	LP2, FP2, LMM2, FMM2, LMG1, FMG2, LMB2, FMB1, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza massa grassa non valida
TC_1.2_6	LP2, FP2, LMM2, FMM2, LMG2, FMG1, LMB2, FMB1, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato massa grassa non valido
TC_1.2_7	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB1, FMB2, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza misurazione braccia non valida
TC_1.2_8	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB1, LMGA2, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato misurazione braccia non valido
TC_1.2_9	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA1, FMGA2, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza misurazione gambe non valida
TC_1.2_10	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA1, LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato misurazione gambe non valido
TC_1.2_11	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2, LMP1, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza misurazione petto non valida

TC_1.2_12	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP1, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato misurazione petto non valido
TC_1.2_13	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF1, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza misurazione fianchi non valida
TC_1.2_14	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF1, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: formato misurazione fianchi non valido
TC_1.2_15	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS1, FMS2, LAA2, FAA2, LPS2, FPS2	Errore: lunghezza misurazione spalle non valida
TC_1.2_16	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS2, FMS1, LAA2, FAA2, LPS2, FPS2	Errore: formato misurazione spalle non valida
TC_1.2_17	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA1, FAA2, LPS2, FPS2	Errore: lunghezza anni di allenamento non valida
TC_1.2_18	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA1, LPS2, FPS2	Errore: formato anni di allenamento non valido
TC_1.2_19	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS1, FPS2	Errore: lunghezza password utente non valida
TC_1.2_20	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2,	Errore: formato password utente non valido



	FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS1	
TC_1.2_21	LP2, FP2, LMM2, FMM2, LMG2, FMG2, LMB2, FMB2, LMGA2, FMGA2. LMP2, FMP2, LMF2, FMF2, LMS2, FMS2, LAA2, FAA2, LPS2, FPS2	Corretto: cambio misurazioni

### 10.1.3. Gestione Account

#### 10.1.3.1. Cambio Email

Parametro: Email utente	
Formato: $\wedge \backslash w + ([ \backslash . - ] ? \backslash w + ) * @ \backslash w + ([ \backslash . - ] ? \backslash w + ) * ( \backslash . \backslash w \{ 2 , \} ) + \$$	
Lunghezza [LE]	<ol style="list-style-type: none"> <li>1. &lt;6 or &gt;40 [error]</li> <li>2. &gt;=6 and &lt;=40 [property LE_OK]</li> </ol>
Formato [FE]	<ol style="list-style-type: none"> <li>1. Non match [if LE_OK] [error]</li> <li>2. Match [if LE_OK] [property FE_OK]</li> </ol>

Parametro: Password utente	
Formato: $\wedge ( ? = . * [ a - z ] ) ( ? = . * [ A - Z ] ) ( ? = . * \backslash d ) ( ? = . * [ @ \$ ! \% * ? \& ] ) [ A - Z a - z \backslash d @ \$ ! \% * ? \& ] \{ 8 , 2 4 \} \$$	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>1. &lt;8 or &gt;24 [error]</li> <li>2. &gt;=8 and &lt;=24 [property LP_OK]</li> </ol>
Formato [FP]	<ol style="list-style-type: none"> <li>1. Non match [if LP_OK] [error]</li> <li>2. Match [if LP_OK] [property FP_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.3.1_1	LE1, FE2, LP2, FP2	Errore: lunghezza email non valida
TC_1.3.1_2	LE2, FE1, LP2, FP2	Errore: formato email non valido
TC_1.3.1_3	LE2, FE2, LP1, FP2	Errore: lunghezza password non valida
TC_1.3.1_4	LE2, FE2, LP2, FP1	Errore: formato password non valido
TC_1.3.1_5	LE2, FE2, LP2, FP2	Corretto: cambio email

### 10.1.3.2. Cambio Password

Parametro: Password attuale utente	
Formato: $^{\wedge}(\?=.*[a-z])(\?=.*[A-Z])(\?=.*\d)(\?=.*[!@\$%\*\&])[A-Za-z\d@\$!%*\&]{8,24}\$$	
Lunghezza [LPA]	<ol style="list-style-type: none"> <li>1. &lt;8 or &gt;24 [error]</li> <li>2. &gt;=8 and &lt;=24 [property LPA_OK]</li> </ol>
Formato [FPA]	<ol style="list-style-type: none"> <li>1. Non match [if LPA_OK] [error]</li> <li>2. Match [if LPA_OK] [property FPA_OK]</li> </ol>

Parametro: Password nuova utente	
Formato: $^{\wedge}(\?=.*[a-z])(\?=.*[A-Z])(\?=.*\d)(\?=.*[!@\$%\*\&])[A-Za-z\d@\$!%*\&]{8,24}\$$	
Lunghezza [LPN]	<ol style="list-style-type: none"> <li>1. &lt;8 or &gt;24 [error]</li> <li>2. &gt;=8 and &lt;=24 [property LPN_OK]</li> </ol>
Formato [FPN]	<ol style="list-style-type: none"> <li>1. Non match [if LPN_OK] [error]</li> <li>2. Match [if LPN_OK] [property FPN_OK]</li> </ol>

Parametro: Conferma Password utente	
Formato: $^{\wedge}(\?=.*[a-z])(\?=.*[A-Z])(\?=.*\d)(\?=.*[!@\$%\*\&])[A-Za-z\d@\$!%*\&]{8,24}\$$	
Lunghezza [CLP]	<ol style="list-style-type: none"> <li>1. &lt;8 or &gt;24 [error]</li> <li>2. &gt;=8 and &lt;=24 [property CLP_OK]</li> </ol>
Formato [CFP]	<ol style="list-style-type: none"> <li>1. Non match [if CLP_OK] [error]</li> <li>2. Match [if CLP_OK] [property CFP_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.3.2_1	LPA1, FPA2, LPN2, FPN2, CLP2, CFP2	Errore: lunghezza password attuale non valida
TC_1.3.2_2	LPA2, FPA1, LPN2, FPN2, CLP2, CFP2	Errore: formato password attuale non valido
TC_1.3.2_3	LPA2, FPA2, LPN1, FPN2, CLP2, CFP2	Errore: lunghezza password nuova non valida
TC_1.3.2_4	LPA2, FPA2, LPN2, FPN1, CLP2, CFP2	Errore: formato password nuova non valido
TC_1.3.2_5	LPA2, FPA2, LPN2, FPN2, CLP1, CFP2	Errore: lunghezza conferma password nuova non valida
TC_1.3.2_6	LPA2, FPA2, LPN2, FPN2, CLP2, CFP1	Errore: formato conferma password nuova non valido
TC_1.3.2_7	LPA2, FPA2, LPN2, FPN2, CLP2, CFP2	Corretto: cambio password



### 10.1.3.3. Cambio Nome e Cognome

Parametro: Nome utente Formato: [A-Za-zÀ-ÿ\s']{2,50}	
Lunghezza [LN]	<ol style="list-style-type: none"> <li>&lt;1 or &gt;50 [error]</li> <li>&gt;=1 and &lt;=50 [property LN_OK]</li> </ol>
Formato [FN]	<ol style="list-style-type: none"> <li>Non match [if LN_OK] [error]</li> <li>Match [if LN_OK] [property FN_OK]</li> </ol>

Parametro: Cognome utente Formato: [A-Za-zÀ-ÿ\s']{2,50}	
Lunghezza [LC]	<ol style="list-style-type: none"> <li>&lt;1 or &gt;50 [error]</li> <li>&gt;=1 and &lt;=50 [property LC_OK]</li> </ol>
Formato [FC]	<ol style="list-style-type: none"> <li>Non match [if LC_OK] [error]</li> <li>Match [if LC_OK] [property FC_OK]</li> </ol>

Parametro: Password utente Formato: ^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,24}\$	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>&lt;8 or &gt;24 [error]</li> <li>&gt;=8 and &lt;=24 [property LP_OK]</li> </ol>
Formato [FP]	<ol style="list-style-type: none"> <li>Non match [if LP_OK] [error]</li> <li>Match [if LP_OK] [property FP_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.3.3_1	LN1, FN2, LC2, FC2, LP2, FP2	Errore: lunghezza nome non valida
TC_1.3.3_2	LN2, FN1, LC2, FC2, LP2, FP2	Errore: formato nome non valido
TC_1.3.3_3	LN2, FN2, LC1, FC2, LP2, FP2	Errore: lunghezza cognome non valida
TC_1.3.3_4	LN2, FN2, LC2, FC1, LP2, FP2	Errore: formato cognome non valido
TC_1.3.3_5	LN2, FN2, LC2, FC2, LP1, FP2	Errore: lunghezza password non valida
TC_1.3.3_6	LN2, FN2, LC2, FC2, LP2, FP1	Errore: formato password non valido
TC_1.3.3_7	LN2, FN2, LC2, FC2, LP2, FP2	Corretto: cambio nome e cognome

#### 10.1.3.4. Cambio Numero

Parametro: Numero di telefono utente Formato: \d{10}	
Lunghezza [LT]	<ol style="list-style-type: none"> <li>&lt;10 or &gt;10 [error]</li> <li>=10 [property LT_OK]</li> </ol>
Formato [FT]	<ol style="list-style-type: none"> <li>Non match [if LT_OK] [error]</li> <li>Match [if LT_OK] [property FT_OK]</li> </ol>

Parametro: Password utente Formato: ^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\$!%*?&])[A-Za-z\d@\$!%*?&]{8,24}\$	
Lunghezza [LP]	<ol style="list-style-type: none"> <li>&lt;8 or &gt;24 [error]</li> <li>&gt;=8 and &lt;=24 [property LP_OK]</li> </ol>
Formato [FP]	<ol style="list-style-type: none"> <li>Non match [if LP_OK] [error]</li> <li>Match [if LP_OK] [property FP_OK]</li> </ol>

Codice	Test Frame	Esito
TC_1.3.4_1	LT1, FT2, LP2, FP2	Errore: lunghezza numero non valida
TC_1.3.4_2	LT2, FT1, LP2, FP2	Errore: formato numero non valido
TC_1.3.4_3	LT2, FT2, LP1, FP2	Errore: lunghezza password non valida
TC_1.3.4_4	LT2, FT2, LP2, FP1	Errore: formato password non valido
TC_1.3.4_5	LT2, FT2, LP2, FP2	Corretto: cambio numero



#### 10.1.3.5. Cambio Sesso

Parametro: Sesso	
Selezionato [SLT]	1. = false [error] 2. = true [property SLT_OK]

Parametro: Password utente	
Formato: $^{(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@\!%\*?&])[A-Za-z\d@!\%*?&]{8,24}}\$$	
Lunghezza [LP]	1. <8 or >24 [error] 2. >=8 and <=24 [property LP_OK]
Formato [FP]	1. Non match[if LP_OK] [error] 2. Match [if LP_OK] [property FP_OK]

Codice	Test Frame	Esito
TC_1.3.5_1	SLT1, LP2, FP2	Errore: sesso non selezionato
TC_1.3.5_2	SLT2, LP1, FP2	Errore: lunghezza password non valida
TC_1.3.5_3	LT2, LP2, FP1	Errore: formato password non valido
TC_1.3.5_4	LT2, LP2, FP2	Corretto: cambio sesso

## 10.2. Gestione Schede di Allenamento

### 10.2.1. Realizzazione Schede di Allenamento

Parametro: Data Inizio	
Controllo giorno corrente [DI]	<ol style="list-style-type: none"> <li>1. Data Inizio &lt; data del giorno corrente [error]</li> <li>2. Data Inizio &gt;= data del giorno corrente [property DI_OK]</li> </ol>

Parametro: Data Fine	
Controllo giorno corrente [DF]	<ol style="list-style-type: none"> <li>1. Data Fine &lt; Data Inizio [error]</li> <li>2. Data Fine &gt;= Data Inizio [property DF_OK]</li> </ol>

Parametro: Titolo Formato: ^\S{2,30}\$	
Lunghezza [LT]	<ol style="list-style-type: none"> <li>1. &lt;2 or &gt;30 [error]</li> <li>2. &gt;=2 and &lt;=30 [property LT_OK]</li> </ol>
Formato [FT]	<ol style="list-style-type: none"> <li>1. Non match [if LT_OK] [error]</li> <li>2. Match [if LT_OK] [property FT_OK]</li> </ol>

Parametro: Nome Esercizio Formato: ^(?=\s*\S)([\w\s]{2,30})\$	
Lunghezza [LE]	<ol style="list-style-type: none"> <li>3. &lt;2 or &gt;30 [error]</li> <li>4. &gt;=2 and &lt;=30 [property LE_OK]</li> </ol>
Formato [FE]	<ol style="list-style-type: none"> <li>3. Non match [if LE_OK] [error]</li> <li>4. Match [if LE_OK] [property FE_OK]</li> </ol>

Parametro: Numero Set Formato: ^[1-9]\d*\$	
Lunghezza [LS]	<ol style="list-style-type: none"> <li>5. &lt;1 [error]</li> <li>6. &gt;=1 [property LS_OK]</li> </ol>
Formato [FS]	<ol style="list-style-type: none"> <li>5. Non match [if LS_OK] [error]</li> <li>6. Match [if LS_OK] [property FS_OK]</li> </ol>

Parametro: Numero ripetizioni Formato: ^[1-9]\d*\$	
Lunghezza [LR]	<ol style="list-style-type: none"> <li>7. &lt;1 [error]</li> <li>8. &gt;=1 [property LR_OK]</li> </ol>
Formato [FR]	<ol style="list-style-type: none"> <li>7. Non match [if LR_OK] [error]</li> <li>8. Match [if LR_OK] [property FR_OK]</li> </ol>

Parametro: Tempo di recupero	
Lunghezza [LTR]	9. <1 [error] 10. >=1 [property LTR_OK]

Codice	Test Frame	Esito
TC_2.1_1	DI1, DF2, LT2, FT2, LE2, FE2, LS2, FS2, LR2, FR2, LTR2	Errore: data inizio precedente alla data corrente
TC_2.1_2	DI2, DF1, LT2, FT2, LE2, FE2, LS2, FS2, LR2, FR2, LTR2	Errore: data fine precedente alla data inizio
TC_2.1_3	DI2, DF2, LT1, FT2, LE2, FE2, LS2, FS2, LR2, FR2, LTR2	Errore: lunghezza titolo non valida
TC_2.1_4	DI2, DF2, LT2, FT1, LE2, FE2, LS2, FS2, LR2, FR2, LTR2	Errore: formato titolo non valido
TC_2.1_5	DI2, DF2, LT2, FT2, LE1, FE2, LS2, FS2, LR2, FR2, LTR2	Errore: lunghezza nome esercizio non valida
TC_2.1_6	DI2, DF2, LT2, FT2, LE2, FE1, LS2, FS2, LR2, FR2, LTR2	Errore: formato nome esercizio non valido
TC_2.1_7	DI2, DF2, LT2, FT2, LE2, FE2, LS1, FS2, LR2, FR2, LTR2	Errore: lunghezza numero set non valida
TC_2.1_8	DI2, DF2, LT2, FT2, LE2, FE2, LS2, FS1, LR2, FR2, LTR2	Errore: formato numero set non valido
TC_2.1_9	DI2, DF2, LT2, FT2, LE2, FE2, LS2, FS2, LR1, FR2, LTR2	Errore: lunghezza numero ripetizioni non valida
TC_2.1_10	DI2, DF2, LT2, FT2, LE2, FE2, LS2, FS2, LR2, FR1, LTR2	Errore: formato numero ripetizioni non valido
TC_2.1_11	DI2, DF2, LT2, FT2, LE2, FE2, LS2, FS2, LR2, FR2, LTR1	Errore: lunghezza tempo di recupero non valida
TC_2.1_12	DI2, DF2, LT2, FT2, LE2, FE2, LS2, FS2, LR2, FR2, LTR2	Corretto: creazione scheda di allenamento



## 11. Scheduling delle attività di testing

---

Le attività di pianificazione del testing saranno programmate seguendo quanto definito nei capitoli precedenti, ovvero immediatamente dopo la fase di progettazione necessaria per la pianificazione stessa.

La stesura dei casi di test sarà effettuata in parallelo con lo sviluppo del codice, consentendo un approccio collaborativo e sincronizzato tra la definizione dei test e l'implementazione del sistema.

L'esecuzione dei test sarà pianificata sia durante il processo di sviluppo che successivamente alla sua implementazione. Una volta completato lo sviluppo, tutti i test saranno ripetuti per garantire il corretto funzionamento del sistema e per generare i report finali che documenteranno l'esito complessivo delle attività di testing.