



Laurea Magistrale in informatica-Università di Salerno
Corso di Gestione dei Progetti Software- Prof.ssa F.Ferrucci



FITDIARY

DECIDE, COMMIT, SUCCEED

TP Test Plan

Riferimento	2021_RAD_C7_FitDiary_V1.0, 2021_SDD_C7_FitDiary_V1.0
Versione	1.0
Data	08/12/2021
Destinatario	Prof.ssa Filomena Ferrucci, Prof.re Fabio Palomba
Presentato da	C07 Team FitDiary: Salvatore Fasano, Gianluca Spinelli, Daniele De Marco, Ilaria De Sio, Rebecca Di Matteo, Daniele Giaquinto, Davide La Gamba, Leonardo Monaco, Simone Spera, Antonio Trapanese.
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
08/12/2021	0.1	Stesura dei capitoli fino all'11 contenenti gli approcci di testing, componenti da testare e non e criteri pass/Fail	Salvatore Fasano, Gianluca Spinelli [Project Managers]
08/12/2021	1.0	TP-Stesura test Cases nel capitolo 12	Tutto il team



Sommario

Revision History	2
1. Introduzione	5
2. Relazione con altri documenti	6
3. Panoramica del sistema	7
4. Features da testare/da non testare	8
5. Pass/Fail criteria	9
6. Approccio	10
6.1 Testing di Sistema	10
6.1.1 Functional testing	10
6.1.2 Performance Testing	10
6.1.3 Pilot Testing	10
6.1.4 Acceptance Testing	11
6.1.5 Installation Testing	11
6.2 Testing di Integrazione	11
6.3 Testing di Unità	13
6.4 Ispezione del Codice	13
7. Sospensione e ripristino	14
7.1 Criteri di sospensione	14
7.2 Criteri di ripristino	14
9. Test deliverables	14
10. Materiale di testing	15
11. Responsabilità	15
12. Test Cases	16
12.1 Gestione Utente	16
12.1.1 Registrazione Preparatore	16
12.1.2 Login	19
12.1.3 Inserimento Nuovo Cliente	20
12.1.4 Inserimento Dati Personali Cliente	21
12.1.5 Modifica Dati Personali (cliente)	23
12.2 Gestione Protocollo	25
12.2.1 Creazione Protocollo	25



12.2.2 Modifica Scheda Allenamento.....	27
12.3 Gestione Report.....	28
12.3.1 Inserimento Report Protocollo.....	28



1. Introduzione

Il sistema che si vuole realizzare ha come obiettivo principale quello di facilitare l'interazione tra i preparatori e i propri clienti e migliorare la gestione dei loro dati. Attraverso una piattaforma online il preparatore potrà creare account per i propri clienti e gestire i loro protocolli alimentari e/o di allenamento. Il sistema proposto permette al preparatore di facilitare la gestione dei propri clienti e dei loro protocolli.

Il documento di Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma FitDiary. Il fine è quello di garantire che ogni aspetto funzioni in modo corretto. 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 una piattaforma priva di malfunzionamenti.

Sono state pianificate attività di testing per i seguenti sottosistemi:

- Gestione Utente
- Gestione Protocollo
- Gestione Report



2. Relazione con altri documenti

Questo documento è correlato a tutti i documenti prodotti fino al rilascio del sistema; quindi, verrà modificato in futuro dopo il rilascio di altri documenti non ancora prodotti. I test case sono basati sulle funzionalità del sistema, individuate e raccolte nei seguenti documenti prodotti:

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.

Relazioni con il Object Design Document (ODD)

Per ciò che concerne i test di unità e di integrazione, maggiormente legati allo ODD e alla divisione in package del sistema, essi saranno scritti e documentati unicamente all'interno del codice dell'applicativo. Per tale motivo, nel presente documento, non vi saranno riferimenti al loro design.



3. Panoramica del sistema

Il sistema proposto basa la sua architettura sul modello Client-Server, in particolare, per quanto riguarda il backend verrà utilizzata la tecnologia Java SpringBoot mentre, per quanto riguarda il frontend, verrà utilizzata la libreria ReactJS. La comunicazione tra backend e frontend avverrà mediante delle REST API che implementate all'interno del backend potranno essere utilizzare per la fruizione dei servizi di business, mediante richieste HTTP.

Per la gestione del database saranno usati:

- Spring JPA per il collegamento al database.
- H2 come database in fase di sviluppo.
- PostgreSQL per il database in production.

Il sistema viene quindi suddiviso nei seguenti sottosistemi:

- **Gestione utenza:** si occupa di gestire le funzioni di login, logout, modifica delle impostazioni utente, l'inserimento e la modifica dei dati personali, la registrazione del preparatore ed inoltre dell'eliminazione degli utenti da parte dell'amministratore.
- **Gestione protocollo:** è responsabile dell'elaborazione, dell'inserimento e della modifica dei protocolli.
- **Gestione abbonamento:** si occupa del canale di comunicazione con il sistema esterno "Stripe" per gestire i pagamenti e le notifiche di abbonamenti in scadenza.
- **Gestione report:** è responsabile della modifica e dell'inserimento dei report.
- **Gestione stima progressi:** è responsabile della stima dei progressi e del calcolo degli stessi tramite un modulo di intelligenza artificiale.
- **Persistenza:** Si occupa della gestione della persistenza tramite l'ausilio di un database.



4. Features da testare/da non testare

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

- Gestione Utente
 - Registrazione Preparatore
 - Login
 - Inserimento Nuovo Cliente
 - Inserimento Dati Personali (cliente)
 - Modifica Dati Personali (cliente)
- Gestione Protocollo
 - Creazione Protocollo
 - Modifica Scheda Allenamento
- Gestione Report
 - Inserimento Report Protocollo

Le funzionalità di cui non si andrà ad effettuare le attività di testing riguardano requisiti funzionali di bassa o media priorità; sono inoltre escluse le funzionalità che non prevedono input manuale da parte dell'utente - ad esempio attività riguardanti esclusivamente visualizzazioni di dati. Il contenuto del sottosistema "Gestione Stima Progressi" che rappresenta il modulo di Intelligenza Artificiale per la previsione dei progressi, verrà testato in futuro con differenti tecniche e sarà quindi trattato in una diversa documentazione.



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 se tutti i seguenti vincoli saranno rispettati:

- Testare tutti i requisiti funzionali ad alta priorità;
- 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%.



6. Approccio

Il testing dell'intero sistema si compone di tre fasi: testing di sistema, testing di integrazione e testing di unità. Verranno progettati nell'ordine appena definito, ma verranno eseguiti in ordine inverso. Prima della fase di implementazione del sistema, avverrà la progettazione dei casi di test di sistema, perfezionati in seguito nella loro fase di esecuzione. Per la progettazione dei test di unità si utilizzerà l'approccio Test-Driven; quindi, prima della fase implementativa avverrà la progettazione dei casi di test di unità. Durante lo sviluppo saranno eseguite periodiche attività di revisione sul codice prodotto. Il testing di sistema è stato pianificato in seguito alla stesura del documento Requirements Analysis Document, mentre la pianificazione del testing di integrazione avverrà dopo la stesura del System Design Document.

6.1 Testing di Sistema

Per il testing di sistema sarà utilizzato il tool Selenium IDE, che permette di registrare le azioni che un utente può intraprendere sul browser, in modo da poter implementare ed eseguire i test case di sistema. Il server, per la fase di testing, verrà deployato in localhost.

6.1.1 Functional testing

Il functional testing ha il fine di validare i requisiti funzionali. Consiste nell'individuare i possibili faults generati dagli input degli utenti. Per cercare di individuare i possibili input che causano fault al sistema ed allo stesso tempo ridurre il costo di testing, verrà utilizzata la tecnica del *Category Partition* per individuare i test case su cui testare il sistema.

6.1.2 Performance Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione del performance testing.

6.1.3 Pilot Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione del pilot testing.



6.1.4 Acceptance Testing

L'acceptance testing verrà effettuato solo sul functional testing, ed il Project Manager simulerà la figura del cliente.

6.1.5 Installation Testing

A causa del basso budget a disposizione, non si assicura l'esecuzione dell'installation testing.

6.2 Testing di Integrazione

Verrà utilizzato un approccio bottom-up, metodo ritenuto più adatto per un software basato sul paradigma Object Oriented. La definizione dei test case avverrà tramite il framework JUnit, mentre verrà usato Mockito per il mocking. Verrà valutato l'utilizzo di Github Actions per realizzare la Continuous Integration, in alternativa a Travis CI. L'automatizzazione del run dei test sarà gestita da Maven, ed infine come tool di misurazione e report coverage sarà utilizzato JaCoCo. Il test di integrazione sarà il medesimo per tutte le componenti da testare. Nello specifico, si procederà prima con il test delle classi Service, e successivamente con il test delle classi Controller. Durante questa seconda esecuzione, la chiamata al controller sarà mockata usando Mockito.

Per ciò che concerne le dipendenze tra i sottosistemi, si riporta di seguito il diagramma architetturale.



GestioneReportController è collegato a GestioneStimaProgressiService per consentire il calcolo della stima al momento dell'inserimento di un nuovo Report.



6.3 Testing di Unità

Per il testing di unità la strategia prevista consiste nel testare ogni metodo delle classi del sistema. Da esse, sono escluse le interfacce e le classi entity, poiché quest'ultime presentano solo metodi getters e setters. I casi di test saranno definiti attraverso un approccio black-box e saranno documentati direttamente nel codice, attraverso l'uso del framework per il testing di classi Java JUnit. Per ogni Production Class sarà definita una Test Class che rispetterà il formato NomeProductionClassTest. Tali classi saranno scritte prima della stesura delle Production class, per eliminare effetti BIAS dovuti alla conoscenza del codice. Le stesse classi saranno poi revisionate e modificate da sviluppatori differenti. Altre tecnologie usate in tale fase saranno:

- Mockito: per la costruzione degli stub e l'isolamento della componente testata.
- JaCoCo: per il calcolo di metriche tra le quali la Branch Coverage.
- Maven: per la build e l'esecuzione automatica dei tests.

6.4 Ispezione del Codice

Per aumentare la qualità del codice ci si affiderà principalmente ai controlli del tool Checkstyle e della CI/CD con Maven. Sebbene vi sia l'intenzione di fare ispezione del codice, anche se non costante e approfondita, a causa del basso budget non si può assicurare tale pratica.



7. Sospensione e ripristino

In questa sezione verranno specificati i criteri di sospensione del test e le attività di test che dovranno essere ripetute quando si riprende il test.

7.1 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.

7.2 Criteri di ripristino

Il testing verrà ripreso dopo aver risolto i fault individuati.

9. Test deliverables

I documenti che saranno prodotti durante questa fase sono i seguenti:

- Test Plan;
- Test Case Specification;
- Test Integration Case Document;
- Test Execution Report;
- Test Incident Report;
- Test Summary Report.



10. Materiale di testing

Le risorse che vengono utilizzate dalle attività di testing comprendono i documenti di progetto Requirement Analysis Document, System Design Document ed Object Design Document, a partire dai quali vengono individuate le componenti da testare, rispettivamente, nel testing di sistema, nel testing di integrazione e nel testing d'unità. L'esecuzione dei test necessita di un server correttamente configurato su cui siano installati Java. La configurazione deve avvenire come da manuale d'installazione.

Il testing è condotto utilizzando alcuni dei framework più famosi ed efficaci in ambienti Java: JUnit, Mockito ed AssertJ. Ad essi viene affiancata tutta la suite di test relativa a Spring framework adottato per la realizzazione della web app. L'utilizzo di JUnit e AssertJ riguarda sia il testing di unità che quello di integrazione, mentre Mockito viene utilizzato solo nel primo caso per “mascherare” le dipendenze. I test sono eseguiti ad ogni modifica apportata al sistema, in un ambiente di Continuous Integration: ciò è possibile grazie all'utilizzo di Maven e di un tool di CI come GitHub Action.

Per Maven sono importanti i seguenti plugin:

- SureFire: utilizzato durante la fase di test di unità, permette la generazione di report in due diversi formati (*.txt e *.xml);
- FailSafe: utilizzato durante la fase di test di integrazione, in caso di fallimento del test permette di farlo in modo sicuro.

11. Responsabilità

I team members verranno suddivisi in due team di sviluppo formati da quattro componenti, rispettivamente per backend e frontend. Ognuno di essi dovrà essere in ogni caso conoscente di tutto il lato del testing del sistema realizzato.



12. Test Cases

12.1 Gestione Utente

12.1.1 Registrazione Preparatore

Parametro: E-mail	
Formato:	
$\wedge[A-Z0-9_ \% + -] + @ [A-Z0-9 _ -] + \wedge . [A-Z] \{ 2, 10 \} \$$	
Nome Categoria	Scelte per la Categoria
Formato [FE]	1. Formato email = false [ERROR] 2. Formato email = true [PROPERTY FE_OK]
Parametro: Password	
Formato:	
$\wedge (? = . * [a - z]) (? = . * [A - Z]) (? = . * \backslash d) (? = . * [@ \$! \% * ? \&]) [A - Z a - z \backslash d @ \$! \% * ? \&] \{ 8, \} \$$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LP]	1. Lunghezza < 8 [ERROR] 2. Lunghezza >= 8 [PROPERTY LP_OK]
Formato [FP]	1. Formato = false [IF LP_OK] [ERROR] 2. Formato = true [IF LP_OK] [PROPERTY FP_OK]
Parametro: Conferma Password	
Formato:	
$\wedge (? = . * [a - z]) (? = . * [A - Z]) (? = . * \backslash d) (? = . * [@ \$! \% * ? \&]) [A - Z a - z \backslash d @ \$! \% * ? \&] \{ 8, \} \$$	
Nome Categoria	Scelte per la Categoria
Match [MP]	1. Match con password = false [error] 2. Match con password = true [PROPERTY MP_OK]
Parametro: Nome	
Formato:	
$\wedge [A - Z] [a - z] ^ *$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	1. Lunghezza < 1 [ERROR] 2. Lunghezza >= 1 [PROPERTY LN_OK]
Formato [FN]	1. Formato = false [IF LN_OK] [ERROR] 2. Formato = true [IF LN_OK] [PROPERTY FN_OK]



Parametro: Cognome	
Formato: $\wedge[A-Z][a-z]^*$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LC]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LC_OK]
Formato [FC]	1. Formato = false [IF LC_OK] [ERROR] 2. Formato= true [IF LC_OK][PROPERTY FN_OK]
Parametro: Sesso	
Nome Categoria	Scelte per la Categoria
Selezionato[SE]	1. Selezionato=0 [ERROR] 2. Selezionato=1 [PROPERTY SE_OK]
Parametro: Data Nascita	
Formato: [0-9]{4}-[0-9]{2}-[0-9]{2}	
Nome Categoria	Scelte per la Categoria
Formato [FDN]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FDN_OK]
Inserimento [IDN]	1. Data corrispondente a data odierna [IF FDN_OK] [ERROR] 2. Data successiva a data odierna [IF FDN_OK] [ERROR] 3. Data precedente a data odierna [PROPERTY IDN_OK]



Test Case ID	Test Frame	Esito
TC_1.1.1	FE1	Errato: formato e-mail errata
TC_1.1.2	FE2, LP1	Errato: lunghezza password non corretta
TC_1.1.3	FE2, LP2, FP1	Errato: formato password non corretto
TC_1.1.4	FE2, LP2, FP2, MP1	Errato: le due password non coincidono
TC_1.1.5	FE2, LP2, FP2, MP2, LN1	Errato: lunghezza nome non corretta
TC_1.1.6	FE2, LP2, FP2, MP2, LN2, FN1	Errato: formato nome non corretto
TC_1.1.7	FE2, LP2, FP2, MP2, LN2, FN2, LC1	Errato: lunghezza cognome non corretta
TC_1.1.8	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC1	Errato: formato cognome non corretto
TC_1.1.9	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC2, SE1	Errato: sesso non selezionato
TC_1.1.10	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC2, SE2, FDN1	Errato: formato data di nascita non corretto
TC_1.1.11	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC2, SE2, FDN2, IDN1	Errato: data corrispondente a data odierna
TC_1.1.12	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC2, SE2, FDN2, IDN2	Errato: data successiva a data odierna
TC_1.1.13	FE2, LP2, FP2, MP2, LN2, FN2, LC2, FC2, SE2, FDN2, IDN3	Corretto



12.1.2 Login

Parametro: E-mail	
Formato:	
$\wedge[A-Z0-9._ \% + -] + @ [A-Z0-9.-] + \. [A-Z] \{2,10\} \$$	
Nome Categoria	Scelte per la Categoria
Match [ME]	1. Match con email = false[error] 2. Match con email = true [PROPERTY ME_OK]
Parametro: Password	
Formato:	
$\wedge (? = . * [a-z]) (? = . * [A-Z]) (? = . * \backslash d) (? = . * [@ \$! \% * ? \&]) [A-Za-z \backslash d @ \$! \% * ? \&] \{ 8 , \} \$$	
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_1.2.1	ME1	Errato: match e-mail errata
TC_1.2.2	ME2, MP1	Errato: match password errato
TC_1.2.3	ME2, MP2	Corretto



12.1.3 Inserimento Nuovo Cliente

Parametro: E-mail	
Formato:	
$\wedge[A-Z0-9._\%+-]+\@[A-Z0-9.-]+\.[A-Z]\{2,10\}\$$	
Nome Categoria	Scelte per la Categoria
Formato [FE]	1. Formato email = false [ERROR] 2. Formato email = true [PROPERTY FE_OK]
Parametro: Nome	
Formato:	
$\wedge[A-Z][a-z]^*$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LN_OK]
Formato [FN]	1. Formato = false [IF LN_OK] [ERROR] 2. Formato= true [IF LN_OK] [PROPERTY FN_OK]
Parametro: Cognome	
Formato:	
$\wedge[A-Z][a-z]^*$	
Nome Categoria	Scelte per la Categoria
Lunghezza [LC]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LC_OK]
Formato [FC]	1. Formato = false [IF LC_OK] [ERROR] 2. Formato= true [IF LC_OK] [PROPERTY FN_OK]

Test Case ID	Test Frame	Esito
TC_1.3.1	FE1	Errato: formato e-mail errata
TC_1.3.2	FE2, LN1	Errato: lunghezza nome errata
TC_1.3.3	FE2, LN2, FN1	Errato: formato nome errato
TC_1.3.4	FE2, LN2, FN2, LC1	Errato: lunghezza cognome errata
TC_1.3.5	FE2, LN2, FN2, LC2, FC1	Errato: formato cognome errato
TC_1.3.6	FE2, LN2, FN2, LC2, FC2	Corretto



12.1.4 Inserimento Dati Personali Cliente

Parametro: Data Nascita	
Formato: [0-9]{4}-[0-9]{2}-[0-9]{2}	
Nome Categoria	Scelte per la Categoria
Formato [FDN]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FDN_OK]
Inserimento [IDN]	1. Data corrispondente a data odierna PROPERTY [IF FDN_OK] [ERROR] 2. Data successiva a data odierna [IF FDN_OK] [ERROR] 3. Data precedente a data odierna [IF FDN_OK] [PROPERTY IDN_OK]
Parametro: Numero Telefono	
Formato: ^[+03][0-9]{3,14}	
Nome Categoria	Scelte per la Categoria
Lunghezza[LNT]	1. Lunghezza<4[ERROR] 2. Lunghezza>15[ERROR] 3. Lunghezza >=4 && Lunghezza <=15 [PROPERTY LNT_OK]
Formato [FNT]	1. Formato = false [IF LNT_OK] [ERROR] 2. Formato= true [IF LNT_OK] [PROPERTY FNT_OK]
Parametro: Città	
Formato: ^[A-Z][A-Z a-z]*	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LN_OK]
Formato[FN]	1. Formato = false [IF LN_OK] [ERROR] 2. Formato= true [IF LN_OK] [PROPERTY FN_OK]
Parametro: Via	
Formato: ^[A-Z][A-Z a-z]*	
Nome Categoria	Scelte per la Categoria
Lunghezza [LV]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LN_OK]
Formato [FV]	1. Formato = false [IF LV_OK] [ERROR] 2. Formato= true [IF LV_OK] [PROPERTY FV_OK]



Parametro: Cap	
Formato: [0-9]{5}	
Nome Categoria	Scelte per la Categoria
Lunghezza [LC]	1. Lunghezza cap <5[ERROR] 2. Lunghezza cap >5 [ERROR] 3. Lunghezza cap =5[PROPERTY LC_OK]

Test Case ID	Test Frame	Esito
TC_1.4.1	FDN1	Errato: formato data errata
TC_1.4.2	FDN2, IDN1	Errato: data corrispondente a data odierna
TC_1.4.3	FDN2, IDN2	Errato: data successiva alla data odierna
TC_1.4.4	FDN2, IDN3, LNT1	Errato: lunghezza numero di telefono errata
TC_1.4.5	FDN2, IDN3, LNT2	Errato: lunghezza numero di telefono errata
TC_1.4.6	FDN2, IDN3, LNT3, FNT1	Errato: formato numero di telefono errato
TC_1.4.7	FDN2, IDN3, LNT3, FNT2, LN1	Errato: lunghezza nome città errata
TC_1.4.8	FDN2, IDN3, LNT3, FNT2, LN2, FN1	Errato: formato nome città errato
TC_1.4.9	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV1	Errato: lunghezza via errata
TC_1.4.10	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV1	Errato: formato via errato
TC_1.4.11	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC1	Errato: lunghezza cap errata
TC_1.4.12	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC2	Errato: lunghezza cap errata
TC_1.4.13	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC3	Corretto



12.1.5 Modifica Dati Personali (cliente)

Parametro: Data Nascita	
Formato: [0-9]{4}-[0-9]{2}-[0-9]{2}	
Nome Categoria	Scelte per la Categoria
Formato [FDN]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FDN_OK]
Inserimento [IDN]	1. Data successiva a data odierna [IF FDN_OK] [ERROR] 2. Data corrispondente a data odierna [IF FDN_OK] [ERROR] 3. Data precedente a data odierna [IF FDN_OK] [PROPERTY IDN_OK]
Parametro: Numero Telefono	
Formato: ^[+03][0-9]{3,14}	
Nome Categoria	Scelte per la Categoria
Lunghezza[LNT]	1. Lunghezza<4[ERROR] 2. Lunghezza>15[ERROR] 3. Lunghezza >=4 && Lunghezza <=15 [PROPERTY LNT_OK]
Formato [FNT]	1. Formato = false [IF LNT_OK] [ERROR] 2. Formato= true [IF LNT_OK] [PROPERTY FNT_OK]
Parametro: Città	
Formato: ^[A-Z][A-Z a-z]*	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LN_OK]
Formato[FN]	1. Formato = false [IF LN_OK] [ERROR] 2. Formato= true [IF LN_OK] [PROPERTY FN_OK]
Parametro: Via	
Formato: ^[A-Z][A-Z a-z]*	
Nome Categoria	Scelte per la Categoria
Lunghezza [LV]	1. Lunghezza <1 [ERROR] 2. Lunghezza >=1 [PROPERTY LN_OK]
Formato [FV]	1. Formato = false [IF LV_OK] [ERROR]



	2. Formato= true [IF LV_OK] [PROPERTY FV_OK]
Parametro: Cap	
Formato: [0-9]{5}	
Nome Categoria	Scelte per la Categoria
Lunghezza [LC]	1. Lunghezza cap <5[ERROR] 2. Lunghezza cap >5[ERROR] 3. Lunghezza cap =5 [PROPERTY LC_OK]

Test Case ID	Test Frame	Esito
TC_1.5.1	FDN1	Errato: formato data errata
TC_1.5.2	FDN2, IDN1	Errato: data successiva a data odierna
TC_1.5.3	FDN2, IDN2	Errato: data corrispondente alla data odierna
TC_1.5.4	FDN2, IDN3, LNT1	Errato: lunghezza numero di telefono errata
TC_1.5.5	FDN2, IDN3, LNT2	Errato: lunghezza numero di telefono errata
TC_1.5.6	FDN2, IDN3, LNT3, FNT1	Errato: formato numero di telefono errato
TC_1.5.7	FDN2, IDN3, LNT3, FNT2, LN1	Errato: lunghezza nome città errata
TC_1.5.8	FDN2, IDN3, LNT3, FNT2, LN2, FN1	Errato: formato nome città errato
TC_1.5.9	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV1	Errato: lunghezza via errata
TC_1.5.10	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV1	Errato: formato via errato
TC_1.5.11	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC1	Errato: lunghezza cap errata
TC_1.5.12	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC2	Errato: lunghezza cap errata
TC_1.5.13	FDN2, IDN3, LNT3, FNT2, LN2, FN2, LV2, FV2, LC3	Corretto



12.2 Gestione Protocollo

12.2.1 Creazione Protocollo

Parametro: Data Scadenza	
Formato: [0-9]{4}-[0-9]{2}-[0-9]{2}	
Nome Categoria	Scelte per la Categoria
Formato [FDS]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FDS_OK]
Formato [IDS]	1. Data precedente a data odierna [IF FDS_OK] [ERROR] 2. Data corrispondente a data odierna [IF FDS_OK] [ERROR] 3. Data successiva a data odierna [IF FDN_OK] [PROPERTY IDN_OK]
Parametro: File Allenamento	
Nome Categoria	Scelte per la Categoria
Estensione[EFALL]	1. Estensione != .csv [ERROR] 2. Estensione =.csv [PROPERTY EFALL_OK]
Grandezza[GFALL]	1. Grandezza<=0 KB[IF EFALL_OK] [ERROR] 2. Grandezza>50 MB [IF EFALL_OK] [ERROR] 3. Grandezza>=0 KB&& Grandezza <= 50 MB [IF EFALL_OK] [PROPERTY GFALL_OK]
Parametro: File Alimentazione	
Nome Categoria	Scelte per la Categoria
Estensione[EFAL]	1. Estensione != .csv [ERROR] 2. Estensione =.csv [PROPERTY EFAL_OK]
Grandezza[GFAL]	1. Grandezza<=0 KB[IF EFAL_OK] [ERROR] 2. Grandezza>50 MB [IF EFAL_OK] [ERROR] 3. Grandezza>=0 KB && Grandezza <= 50 MB [IF EFAL_OK] [PROPERTY GFAL_OK]



Test Case ID	Test Frame	Esito
TC_2.1.1	FDS1	Errato: formato data scadenza errato
TC_2.1.2	FDS2, IDS1	Errato: data precedente a data odierna
TC_2.1.3	FDS2, IDS2	Errato: data corrispondente alla data odierna
TC_2.1.4	FDS2, IDS3, EFALL1	Errato: estensione File Allenamento errata
TC_2.1.5	FDS2, IDS3, EFALL2, GFALL1	Errato: grandezza File Allenamento errata
TC_2.1.6	FDS2, IDS3, EFALL2, GFALL2	Errato: grandezza File Allenamento errata
TC_2.1.7	FDS2, IDS3, EFALL2, GFALL3, EFAL1	Errato: estensione File Alimentazione errata
TC_2.1.8	FDS2, IDS3, EFALL2, GFALL3, EFAL2, GFAL1	Errato: grandezza File Alimentazione errata
TC_2.1.9	FDS2, IDS3, EFALL2, GFALL3, EFAL2, GFAL2	Errato: grandezza File Alimentazione errata
TC_2.1.10	FDS2, IDS3, EFALL2, GFALL3, EFAL2, GFAL3	Corretto



12.2.2 Modifica Scheda Allenamento

Parametro: File Allenamento	
Nome Categoria	Scelte per la Categoria
Estensione[EFALL]	1. Estensione != .csv [ERROR] 2. Estensione =.csv [PROPERTY EFALL_OK]
Grandezza[GFALL]	1. Grandezza<=0 KB [IF EFALL_OK] [ERROR] 2. Grandezza>50 MB [IF EFALL_OK] [ERROR] 3. Grandezza >=0 KB && Grandezza <= 50 MB [IF EFALL_OK] [PROPERTY GFALL_OK]

Test Case ID	Test Frame	Esito
TC_2.2.1	EFALL1	Errato: estensione errata
TC_2.2.2	EFALL2, GFALL1	Errato: grandezza errata
TC_2.2.3	EFALL2, GFALL2	Errato: grandezza errata
TC_2.2.4	EFALL2, GFALL3	Corretto



12.3 Gestione Report

12.3.1 Inserimento Report Protocollo

Parametro: Peso	
Formato: $^{\wedge}[1-9][0-9]^*.[0-9]\{2\}$	
Nome Categoria	Scelte per la Categoria
Formato[FIRP]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FIRP_OK]
Parametro: Circonferenza Braccia	
Formato: $^{\wedge}[1-9][0-9]^*.[0-9]\{2\}$	
Nome Categoria	Scelte per la Categoria
Formato[FCB]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FCB_OK]
Parametro: Circonferenza Addome	
Formato: $^{\wedge}[1-9][0-9]^*.[0-9]\{2\}$	
Nome Categoria	Scelte per la Categoria
Formato[FCA]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FCA_OK]
Parametro: Circonferenza Quadricipite	
Formato: $^{\wedge}[1-9][0-9]^*.[0-9]\{2\}$	
Nome Categoria	Scelte per la Categoria
Formato[FCQ]	1. Formato = false [ERROR] 2. Formato= true [PROPERTY FCQ_OK]
Parametro: Immagine	
Nome Categoria	Scelte per la Categoria
Tipo[TI]	1. Tipo != immagine [ERROR] 2. Tipo=immagine [PROPERTY EI_OK]
Grandezza[GI]	1. Grandezza<=0 KB[IF TI_OK] [ERROR] 2. Grandezza>50 MB [IF TI_OK] [ERROR] 3. Grandezza >=0 KB && Grandezza <=50 MB [IF TI_OK] [PROPERTY GI_OK]



Test Case ID	Test Frame	Esito
TC_3.1.1	FIRP1	Errato: formato peso errato
TC_3.1.2	FIRP2, FCB1	Errato: formato circonferenza braccia errato
TC_3.1.3	FIRP2, FCB2, FCA1	Errato: formato circonferenza addome errato
TC_3.1.4	FIRP2, FCB2, FCA2, FCQ1	Errato: formato circonferenza quadricipite errato
TC_3.1.5	FIRP2, FCB2, FCA2, FCQ2, TI1	Errato: formato file non valido
TC_3.1.6	FIRP2, FCB2, FCA2, FCQ2, TI2, GI1	Errato: grandezza immagine errata
TC_3.1.7	FIRP2, FCB2, FCA2, FCQ2, TI2, GI2	Errato: grandezza immagine errata
TC_3.1.8	FIRP2, FCB2, FCA2, FCQ2, TI2, GI3	Corretto