



# TP – Test Plan

## Progetto TalkAID

Riferimento	C16_TP_ver.1.5
Versione	1.5
Data	<b>19/01/2024</b>
Destinatario	Dipartimento di Informatica dell'Università degli studi di Salerno
Presentato da	C16 Team Member
Approvato da	Carminie Pastore e Nicola Laurino



## Revision History

Data	Versione	Descrizione	Autori
03/12/2023	1.0	Inizio Stesura	Luigi Salvatore Pio Petrillo, Cristian Porzio, Anna Benedetta Salerno
05/12/2023	1.1	Aggiunti Test Cases tabellari	Cristian Porzio, Anna Benedetta Salerno
06/12/2023	1.2	Accorgimenti sul documento	Cristian Porzio
07/12/2023	1.2	Revisione e correzione	Raffaele Monti
09/12/2023	1.3	Revisione pre-consegna	Anna Benedetta Salerno, Cristian Porzio
20/12/2023	1.4	Rimosso Typo	Cristian Porzio
19/01/2024	1.5	Revisione completa	Tutti i Member



## Team Composition

Ruolo	Nome	Posizione	Contatti
Top Manager	Filomena Ferrucci	Rappresentant e del cliente	<a href="mailto:fferrucci@unisa.it">fferrucci@unisa.it</a>
Top Manager	Fabio Palomba	Rappresentant e del cliente	<a href="mailto:fpalomba+is@unisa.it">fpalomba+is@unisa.it</a>
Project Manager	Nicola Laurino	Project Manager	<a href="mailto:n.laurino1@studenti.unisa.it">n.laurino1@studenti.unisa.it</a>
Project Manager	Carmine Pastore	Project Manager	<a href="mailto:c.pastore35@studenti.unisa.it">c.pastore35@studenti.unisa.it</a>
Team Member	Michele D'Arienzo	Team Member	<a href="mailto:m.dariento20@studenti.unisa.it">m.dariento20@studenti.unisa.it</a>
Team Member	Raffaele Monti	Team Member	<a href="mailto:r.monti2@studenti.unisa.it">r.monti2@studenti.unisa.it</a>
Team Member	Cristian Porzio	Team Member	<a href="mailto:c.porzio3@studenti.unisa.it">c.porzio3@studenti.unisa.it</a>
Team Member	Luigi Salvatore Pio Petrillo	Team Member	<a href="mailto:l.petrillo6@studenti.unisa.it">l.petrillo6@studenti.unisa.it</a>
Team Member	Anna Benedetta Salerno	Team Member	<a href="mailto:a.salerno45@studenti.unisa.it">a.salerno45@studenti.unisa.it</a>
Team Member	Samuele Sparno	Team Member	<a href="mailto:s.sparno@studenti.unisa.it">s.sparno@studenti.unisa.it</a>



## Sommario

1. Introduzione .....	5
2. Documenti Correlati .....	5
2.1 Relazione con il documento di analisi .....	5
2.2 Relazione con il System Design Document.....	5
2.3 Relazione con l'Object Design Document .....	5
3. Panoramica del Sistema .....	6
4. Funzionalità da testare .....	7
4.1 Scelta dei test di unità.....	7
4.2 Scelta dei test di sistema.....	7
5. Criteri di Pass/Fail.....	8
6. Approccio .....	8
6.1 Testing di unità.....	8
6.2 Testing di sistema.....	9
Ispezione del Codice.....	9
7. Sospensione e ripresa.....	10
7.1 Criteri di sospensione.....	10
7.2 Criteri di Ripresa .....	10
7.3 Criteri di Terminazione.....	10
8. Materiale per il testing.....	10
9. Test cases .....	11
10. Testing Schedule.....	19



## 1. Introduzione

---

TalkAID si propone di semplificare ed innovare il processo di cura ed esercitazione dei pazienti affetti da problematiche del parlato offrendo ai logopedisti uno strumento per una rapida analisi del progresso dei propri pazienti ed uno strumento efficiente per essere sempre a contatto con loro, integrando una messaggistica privata tra le parti e la possibilità di gestire la propria agenda delle visite mediche.

Il documento Test Plan ha l'obiettivo di descrivere ed analizzare le attività di Testing per la piattaforma TalkAID. Il fine è quello di garantire che ogni aspetto funzioni in maniera corretta.

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 che possano minare l'utilizzabilità del sistema. Sono pianificate attività di testing attraverso input e casi di test specifici per mettere alla prova le funzionalità offerte dal sistema stesso. I risultati dei test saranno fondamentali per individuare le aree in cui intervenire per rimuovere eventuali fault.

## 2. Documenti Correlati

---

### **2.1 Relazione con il documento di analisi**

La progettazione dei casi di test è effettuata tenendo in considerazione le specifiche del sistema, che sono trattate all'interno degli Use Case e gli Scenari presenti nel documento di analisi (RAD).

### **2.2 Relazione con il System Design Document**

I test case contenuti nel Test Plan devono rispettare la suddivisione in sottosistemi presente nel System Design Document(SDD).

### **2.3 Relazione con l'Object Design Document**

Per quanto concerne i test di Unità, sono stati testati i seguenti sottosistemi individuati nell'Object Design Document: "Registration", "Login", "License", "Exercise", "Encryption", "Schedule". Come paradigma per eseguire il testing di tali sottosistemi, si è adottato il Category Partition.



### 3. Panoramica del Sistema

---

Il sistema proposto basa la sua architettura sul modello Three-Tier, sfruttando la compartimentalizzazione delle componenti per uno sviluppo modulare.

Verranno usati HTML5, CSS3 e JavaScript per la parte Front-End e la generazione delle View.

Per la logica applicativa ed il Back-End sarà utilizzato Java.

Per la gestione del Database saranno usati:

- Java e JDBC per il collegamento al Database
- MySQL come Database

Sono stati individuati i seguenti sottosistemi dalle analisi precedenti:

- Registration
- Login
- License
- Encryption
- Exercise
- User
- Schedule
- Condition
- Email
- Message
- Telemetry



## 4. Funzionalità da testare

---

La miglior scelta di testing per qualsiasi sistema software è quello di testare ogni componente del sistema, però ciò non è fattibile per i vincoli imposti e l'ammontare di ore lavorative a disposizione. Il team si occuperà di testare esclusivamente le porzioni di sistema accessibili agli utenti, ovvero dove è previsto input manuale ed i sottosistemi ad essi correlati, al fine di garantire la robustezza delle azioni chiave del progetto individuate attraverso l'alta priorità. Non sono previsti test riguardanti la visualizzazione dei dati.

### 4.1 Scelta dei test di unità

Sottosistema	Funzionalità
Registration	Primo accesso
Login	Accesso al Sistema
License	Validità PIN / Licenza
Encryption	Criptazione Password
Exercise	Conversione file Audio
Schedule	Modifica agenda

È risultato opportuno testare alcune funzionalità cruciali di Back-End, dunque non visualizzabili in Front-end per loro natura, in quanto i loro risultati o malfunzionamenti non sono individuabili con dei test di sistema.

### 4.2 Scelta dei test di sistema

Sottosistema	Funzionalità
Registration	Primo accesso
Login	Accesso al Sistema
Login	Reset Password
User	Modifica dati personali
Message	Invio messaggi
Exercise	Esecuzione di un Esercizio

Non verranno testate:

- Sicurezza.
- Performance.



## 5. Criteri di Pass/Fail

---

L'approccio utilizzato per testare il sistema sarà di tipo funzionale. L'obiettivo che ci poniamo è quello di individuare dei fault durante le fasi di sviluppo in modo che, una volta rilasciato il software, esso contenga quanti meno fault possibili per lo scopo per cui sono stato prodotto.

L'approccio ci incita a provare e riprovare una funzionalità fino a risultare valido in tutte le casistiche utilizzate nel progetto.

Pertanto, il test viene marcato come "PASS" se il comportamento osservato è diverso da quello atteso. In questo caso verrà analizzata la causa dell'errore e verrà risolto. Il test viene invece marcato come "FAILED" nel caso in cui non vengano scovate discordanze dal comportamento atteso.

Tutto il testing sarà considerato valido se i seguenti vincoli saranno rispettati:

- Testare una funzionalità attraverso testing di unità per ogni membro del team;
- Testare una funzionalità attraverso testing di sistema per ogni membro del team;

## 6. Approccio

---

Le attività da svolgere per realizzare il testing sono tre. Lo svolgimento seguirà una strategia bottom-up nelle seguenti fasi: test di sistema e test di unità. Questi saranno progettati nell'ordine appena definito. La loro esecuzione avverrà invece in ordine inverso.

Testeremo parzialmente il sistema per verificare che le caratteristiche da considerarsi cruciali richieste dal committente siano rispettate. Infine, ci occuperemo di individuare gli errori sulle singole componenti da ritenersi cruciali.

### 6.1 Testing di unità

È previsto un rapporto 1:1 tra le classi del sistema sviluppate e le classi di test che si occuperanno di verificarne l'operatività. Verrà testata per ogni classe un suo metodo attraverso svariati input affinché vi sia un'alta copertura delle sue ramificazioni (branches). Se ritenuto necessario, il metodo sarà revisionato se risulta difficile da testare.





Le tecnologie utilizzate in questa fase sono:

- JaCoCo: per il calcolo delle metriche del Branch Coverage del codice.
- Maven: per la build ed esecuzione automatica dei test del progetto.
- JUnit: per la moltitudine di metodi ed annotazioni utili per l'esecuzione dei test.
- SonarCloud: per la Continuous Integration, sfrutterà le tecnologie sopra citate per fornire dopo ogni push, attraverso una UI moderna, l'analisi quali qualità del codice, bug, security issues e coverage.

## 6.2 Testing di sistema

Adotteremo Selenium IDE, un tool che consente di catturare le interazioni dell'utente nel browser per il testing di sistema. Questi dati catturati saranno poi utilizzati per creare ed eseguire i test case del sistema. L'obiettivo di questo approccio di test è verificare i requisiti funzionali, scenari e gli Use Case specificati nei documenti precedenti. Durante la fase di test, lo strumento verrà eseguito in un ambiente locale.

## Ispezione del Codice

Attraverso la CI/CD di SonarCloud si otterrà un riepilogo dettagliato della duplicazione del codice e della sua qualità. Anche se il rapporto è di alto livello e con regole serrate, a causa del tempo a disposizione non è possibile perseguire gli obiettivi preposti per un coding di eccellente qualità.



## 7. Sospensione e ripresa

---

### **7.1 Criteri di sospensione**

Il processo di testing non prevede la possibilità di sospensione totale, anche in presenza di eventuali anomalie rilevate. La sospensione temporanea del testing può avvenire manualmente nel caso in cui si verifichi un errore nella definizione di uno dei test durante l'esecuzione.

### **7.2 Criteri di Ripresa**

Dato l'approccio incrementale adottato nello sviluppo del software, la fase di testing potrebbe essere riattivata con l'introduzione di modifiche o nuove componenti. In tale contesto, è necessario eseguire i test sui nuovi elementi introdotti e, mediante il regression testing, verificare anche quelli già precedentemente testati.

### **7.3 Criteri di Terminazione**

Il test si considera concluso nel momento in cui saranno testati e funzionanti un metodo per ciascuno dei 6 sottosistemi selezionati.

## 8. Materiale per il testing

---

L'esecuzione dei test di sistema necessita di un ambiente desktop in cui siano installati Java SE 11 o successivi, una connessione ad Internet per il collegamento al Database Remoto, Tomcat ed i tool per la suite di test Selenium IDE.

Per l'esecuzione dei restanti test, in locale sarà utilizzato Java 17, JUnit, JaCoCo e Mockito. In remoto, attraverso la CI/CD SonarCloud seguirà la configurazione presente nel pom.xml e nell'action di GitHub ed eseguirà i test in un ambiente con Java 17 sfruttando i report di JaCoCo.



## 9. Test cases

In questa sezione, per ogni sottosistema verranno mostrate le funzionalità che verranno testate. Ogni funzionalità avrà una tabella per ognuno dei suoi parametri, che conterrà i vincoli affinché il valore dell'input sia valido. Per ciascun sottosistema, vi sarà una tabella riepilogativa contenente i test che saranno sviluppati nel Test Case Specification.

### Test di Sistema

#### 9.1.1 Registrazione – Primo accesso

Parametro: PIN	
Nome Categoria	Scelte per la Categoria
Lunghezza [LNP1]	1. Lunghezza = 8 = false [error] 2. Lunghezza = 4 = false [error] 3. Lunghezza = 4 OR = 8 = true [PROPERTY LNP1_OK]
Match [MP1]	1. Presente nel db = false [error] 2. Presente nel db = true [PROPERTY MP1_OK]
Parametro: Email	
FORMATO: $\wedge[\text{a-zA-Z0-9}.\_ \% + - ] + @ [\text{a-zA-Z0-9} . - ] + \. [\text{a-zA-Z} ] \{ 2 , \} \$$	
Nome Categoria	Scelte per la Categoria
Formato [FE1]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE1_OK]
Parametro: Password	
FORMATO: $\wedge ( ? = . * [ \text{A-Z} ] ) ( ? = . * [ \text{a-z} ] ) ( ? = . * [ \text{d} ] ) ( ? = . * [ @ \$ ! \% * ? \& ] ) [ \text{A-Za-z} \backslash \text{d} @ \$ ! \% * ? \& ] \{ 12 , \} \$$	
Formato [FP1]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP1_OK]



Test Case ID	Test Frame	Esito
TC_1.1.1_1	FE1	Errato: email non valida
TC_1.1.1_2	FP1	Errato: password non valida
TC_1.1.1_3	LNP1	Errato: PIN non valido
TC_1.1.1_4	MP1	Errato: PIN non presente
TC_1.1.1_5	LNP1, MP1, FE1, FP1	Errato: nessun vincolo corretto
TC_1.1.1_6	LNP1, MP1, FE1, FP1	Corretto

### 9.1.2 Registrazione – Invito Paziente

Parametro: Nome	
Nome Categoria	Scelte per la Categoria
Lunghezza [LNN]	1. Lunghezza > 1 = false [error] 2. Lunghezza > 1 = true [PROPERTY LNN_OK]
Parametro: Cognome	
Nome Categoria	Scelte per la Categoria
Lunghezza [LNP2]	1. Lunghezza > 1 = false [error] 2. Lunghezza > 1 = true [PROPERTY LNP2_OK]
Parametro: Email	
FORMATO:	
^[a-zA-Z0-9._%+~]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$	
Nome Categoria	Scelte per la Categoria
Formato [FE2]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE2_OK]



Test Case ID	Test Frame	Esito
TC_1.1.2_1	LNN, LNP2, FE2	Errato: nessun vincolo corretto
TC_1.1.2_2	FE2	Errato: email non valida
TC_1.1.2_3	LNN	Errato: nome non valido
TC_1.1.2_4	LNP2	Errato: cognome non valido
TC_1.1.2_5	LNN, LNP2, FE2	Corretto

### 9.2.1 Login – Accesso al Sistema

Parametro: Email	
FORMATO:	
^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$	
Nome Categoria	Scelte per la Categoria
Formato [FE3]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE3_OK]
Match [ME2]	1. Presente nel db = false [error] 2. Presente nel db = true [PROPERTY ME2_OK]
Parametro: Password	
FORMATO:	
^.++\$	
Formato [FP2]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP2_OK]
Match [MP2]	1. Coincide nel db = false [error] 2. Coincide nel db = true [PROPERTY MP2_OK]

Test Case ID	Test Frame	Esito
TC_1.2.1_1	FE3, ME2	Errato: email non presente
TC_1.2.1_2	FP2, MP2	Errato: password non valida
TC_1.2.1_3	FE3	Errata: email non valida
TC_1.2.1_4	FP2, MP2	Errata: password errata
TC_1.2.1_5	FE3, ME2, FP2, MP2	Corretto



### 9.2.2 Login – Recupero Password

Parametro: Email	
FORMATO:	
$\wedge[\text{a-zA-Z0-9}.\_ \% + - ] + @ [\text{a-zA-Z0-9}.\_ \wedge . [\text{a-zA-Z}]\{2,\} \$$	
Nome Categoria	Scelte per la Categoria
Formato [FE4]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE4_OK]
Match [ME3]	1. Presente nel db = false [error] 2. Presente nel db = true [PROPERTY ME3_OK]
Parametro: Codice di Verifica	
Nome Categoria	Scelte per la Categoria
Lunghezza [LN]	1. Lunghezza = 4 = false [error] 2. Lunghezza = 4 = true [PROPERTY LN_OK]
Match [MCDV]	1. Presente nel sistema = false [error] 2. Presente nel sistema = true [PROPERTY MCDV_OK]
Parametro: Password	
FORMATO:	
$\wedge ( ? = . * [ \text{A-Z} ] ) ( ? = . * [ \text{a-z} ] ) ( ? = . * [ \text{d} ] ) ( ? = . * [ @ \$ ! \% * ? \& ] ) [ \text{A-Za-z} \wedge \text{d} @ \$ ! \% * ? \& ] \{ 12, \} \$$	
Formato [FP3]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FP3_OK]
Parametro: Password Ripetuta	
Nome Categoria	Scelta per la Categoria
Match [MPR]	1. Match con password = false [error] 2. Match con password = true [PROPERTY MPR_OK]



Test Case ID	Test Frame	Esito
TC_1.2.2_1	FE4	Errato: email non valida
TC_1.2.2_2	ME3	Errato: email non presente
TC_1.2.2_3	LN	Errato: codice di verifica non corretto
TC_1.2.2_4	MCDV	Errato: codice non presente
TC_1.2.2_5	FP3	Errato: password non valida
TC_1.2.2_6	MPR	Errato: le password non coincidono
TC_1.2.2_7	FE4, ME3, LN, MCDV, FP3, MPR	Corretto

### 9.3 Gestione Profilo Paziente – Aggiunta Patologia

Parametro: Patologia	
Nome Categoria	Scelta per la Categoria
Match [MP3]	1. Presente nel db = false [error] 2. Presente nel db = true [PROPERTY MP3_OK]
Parametro: Gravità	
Nome Categoria	Scelta per la Categoria
Formato [FG]	1. Intero positivo && Valore > 1 && Valore < 10 = true [PROPERTY FG_OK] 2. Intero positivo && Valore > 1 && Valore < 10 = false [error]

Test Case ID	Test Frame	Esito
TC_1.3.1_1	MP3, FG	Errato: vincoli non rispettati
TC_1.3.1_2	FG	Errato: gravità non valida
TC_1.3.1_3	MP3	Errato: patologia non presente
TC_1.3.1_4	MP3	Errato: patologia non valida
TC_1.3.1_4	MP3, FG	Corretto



#### 9.4 Dati Personali – Modifica Dati

Parametro: Nome	
Nome Categoria	Scelta per la Categoria
Formato [FN]	1. Nome = IsNotNull = false [error] 2. Nome = IsNotNull = true [PROPERTY FN_OK]
Parametro: Cognome	
Nome Categoria	Scelta per la Categoria
Formato [FC]	1. Nome = IsNotNull = false [error] 2. Nome = IsNotNull = true [PROPERTY FC_OK]
Parametro: Codice Fiscale	
FORMATO:	
$^{\wedge}[A-Z]\{6\}\backslash d\{8\}[A-EHLMRST]\backslash d\{2\}[A-Z]\backslash d\{3\}[A-Z]\$$	
Formato [FCF]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FCF_OK]
Parametro: Email	
FORMATO:	
$^{\wedge}[a-zA-Z0-9._\%+-]+\@[a-zA-Z0-9.-]+\.[a-zA-Z]\{2,\}\$$	
Nome Categoria	Scelte per la Categoria
Formato [FE5]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FE5_OK]
Parametro: Data di Nascita	
FORMATO:	
$^{\wedge}\backslash d\{4\}-(0[1-9]   1[0-2])-(0[1-9]   [12][0-9]   3[01])\$$	
Nome Categoria	Scelta per la Categoria
Formato [FDDN]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FDDN_OK]
Correttezza [CDDN]	1. Ha almeno 8 anni? = false [error] 2. Ha almeno 8 anni? = true [PROPERTY CDDN_OK]
Parametro: Genere	
Nome Categoria	Scelta per la Categoria
Correttezza [CG]	1. È 'M', 'F' od 'Altro'? = false [error] 2. È 'M', 'F' od 'Altro'? = true [PROPERTY CG_OK]





Parametro: Orario Notifiche	
Nome Categoria	Scelta per la Categoria
Correttezza [CON]	1. È un orario valido espresso in 24h? = false [error] 2. È un orario valido espresso in 24h? = true [PROPERTY CON_OK]
Parametro: Numero di Telefono	
FORMATO: ^\d{10}\$	
Nome Categoria	Scelta per la Categoria
Formato [FNMT]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FNMT_OK]

Test Case ID	Test Frame	Esito
TC_1.4.1_1	FN, FC, FCF, FE5, FDDN, CDDN, CG, CON, FNMT	Errato: nessun dato valido
TC_1.4.1_2	FE5	Errato: email non valida
TC_1.4.1_3	FP1	Errato: password non valida
TC_1.4.1_4	FNMT	Errato: telefono non valido
TC_1.4.1_5	FCF	Errato: codice fiscale non valido
TC_1.4.1_6	FC	Errato: cognome non valido
TC_1.4.1_7	FN	Errato: nome non valido
TC_1.4.1_8	FP1	Errato: password non valida
TC_1.4.1_9	FN, FC, FCF, FE5, FDDN, CDDN, CG, CON, FNMT	Corretto



### 9.5 Prenotazione – Aggiungere orario di visita da parte del Logopedista

Parametro: Data	
Nome Categoria	Scelta per la Categoria
Correttezza [CD]	1. Data >= DataCorrente = true [PROPERTY CD_OK] 2. Data <= DataCorrente = false [error]
FORMATO:	
^\d{4}-(0[1-9]   1[0-2])-(0[1-9]   [12][0-9]   3[01])\$	
Formato [FD]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FD_OK]
Parametro: Fascia Oraria	
Nome Categoria	Scelta per la Categoria
Correttezza [CFO]	1. OrarioInizio <= OrarioFine = true [PROPERTY CFO_OK] 2. OrarioInizio >= Orario Corrente = false [error]

Test Case ID	Test Frame	Esito
TC_1.5.1_1	CD	Errato: data non valida
TC_1.5.1_2	CFO	Errato: orario non valido
TC_1.5.1_3	CD, CFO	Corretto

### 9.6 Messaggi – Invio Messaggio

Parametro: Corpo	
FORMATO:	
^[a-zA-Z0-9\s\.,:;!\"'\\-\\(\\)]+\$	
Nome Categoria	Scelta per la Categoria
Formato [FC]	1. Rispetta il formato = false [error] 2. Rispetta il formato = true [PROPERTY FT_OK]

Test Case ID	Test Frame	Esito
TC_1.6.1_1	FC	Errato: campo vuoto
TC_1.6.1_2	FC	Corretto



## 10. Testing Schedule

---

L'implementazione dei casi di test avverrà quando sarà ritenuto valido e sufficiente il set di metodi sviluppato per una data funzionalità e non in maniera contemporanea al codice.

L'esecuzione dei test avverrà ad ogni build del progetto, sia che questa avvenga in locale, consultabile col riepilogo di JaCoCo sia con il tool di CI/CD implementato.