



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



DCR – Documento Criteri Rispettati Progetto TalkAID

Riferimento	C16_DCR_ver.1.0
Versione	1.0
Data	23/01/2024
Destinatario	Dipartimento di Informatica dell'Università degli studi di Salerno
Presentato da	C16 Team Member
Approvato da	Carmine Pastore e Nicola Laurino



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

Revision History

Data	Versione	Descrizione	Autori
23/01/2024	1.0	Prima stesura	Tutti i Membri

Team Composition

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



Sommario

1. Introduzione	4
1.1 Obiettivo del sistema	4
1.2 Scopo del documento	4
1.3 Riferimenti	4
2. Requisiti Funzionali Implementati.....	5
3. Vincoli collaborativi e comunicativi	5
4. Vincoli tecnici.....	6
5. Criteri di accettazione	6
6. Criteri di premialità	7



1. Introduzione

1.1 Obiettivo del sistema

Il progetto TalkAID rappresenta un passo avanti significativo nel campo della riabilitazione e del supporto alle persone con disturbi del linguaggio. La possibilità di offrire trattamenti completamente a distanza e in maniera asincrona è un'innovazione che potrebbe aprire nuove opportunità per un numero ancora maggiore di individui.

La componente di Intelligenza Artificiale è ritenuta fondamentale ai nostri obiettivi siccome aggiunge un livello di personalizzazione e adattabilità, permettendo ai pazienti di ricevere esercizi mirati in base ai loro errori e difficoltà. Questo non solo rende il trattamento più efficace, ma anche più coinvolgente per i pazienti, incoraggiandoli a perseguire con impegno il percorso di miglioramento.

1.2 Scopo del documento

Il documento ha lo scopo di riassumere i criteri di accettazione, di premialità e mostrare i vincoli progettuali rispettati dai membri del team.

1.3 Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

- [Statement of Work](#)
- [Business Case](#)
- [Requirements Analysis Document](#)
- [System Design Document](#)
- [TestPlan](#)
- [Matrice di Tracciabilità](#)
- [Manuale di Installazione](#)
- [Manuale Utente](#)
- [Test Incident Report](#)
- [Test Summary Report](#)

Di seguito il link al GitHub di TalkAID

- [GitHub](#)



2. Requisiti Funzionali Implementati

Sono qui riportati in forma tabellare, i vari requisiti funzionali implementati, divisi per priorità.

Priorità	Numero Totale	Numero implementati
Elevata	19	19
Media	1	1
Bassa	4	0

3. Vincoli collaborativi e comunicativi

Limitazioni su metodi e strumenti di comunicazione e collaborazione da utilizzare nel progetto:

Criterio	Rispettato	Note
Rispetto delle scadenze	Sì	
Budget effort non superiore a 50 ore per ogni membro del team	Sì	È possibile visionare le ore di ogni singolo membro in questo documento.
Uso di sistemi di versioning	Sì	È stato utilizzato GitHub.
Uso di tool per la gestione di task e attività	Sì	È stato utilizzato Trello.
Uso di un tool di comunicazione tracciabile	No	È stato utilizzato Discord.



4. Vincoli tecnici

Limitazioni sugli artefatti da realizzare:

Criterio	Rispettato	Note
Specifica di minimo 2 e massimo 4 scenari per ogni membro del team	Sì	Sono stati realizzati 18 scenari.
Specifica di minimo 2 e massimo 4 requisiti funzionali e non funzionali per ogni membro del team	Sì	Sono stati specificati 24 Requisiti Funzionali e 17 Requisiti non Funzionali.
Esattamente uno use case per ogni membro del team	Sì	
Esattamente un sequence diagram ogni due membri del team	Sì	
Esattamente un diagramma a scelta tra statechart e activity diagram ogni due membri del team	Sì	Sono stati realizzati 2 Statechart Diagram e un Activity Diagram.
Specifica di un class diagram per team	Sì	
Specifica di minimo 2 e massimo 4 design goal per ogni membro del team	Sì	Sono stati realizzati 18 Design Goals.
Definizione di un diagramma di decomposizione dei sottosistemi per team	Sì	
Definizione di un deployment diagram per team	Sì	
Uso di minimo uno e massimo due design pattern per team	Sì	I Design Pattern scelti sono stati Adapter e Façade.
Uso di UML	Sì	
Ogni membro del team dovrà effettuare il testing di unità, tramite category partition, di esattamente un metodo di una classe sviluppata	Sì	
Ogni membro del team dovrà effettuare il testing di sistema, tramite category partition, di esattamente una funzionalità del sistema sviluppato	Sì	

5. Criteri di accettazione

Criteri che, se non rispettati, portano al fallimento del progetto:



Criterio	Rispettato	Note
Utilizzo appropriato di GitHub	Sì	link
Adeguate utilizzo del pull-based development	Sì	Lista pull-requests
Adeguate utilizzo di Slack	No	Il sistema è diventato a pagamento, abbiamo utilizzato Discord al suo posto.
Adeguate utilizzo di Trello	Sì	
Documentazione adeguata	Sì	
Appropriato test di unità di un metodo sviluppato	Sì	Ogni membro ha eseguito il test di unità di un metodo sviluppato.
Appropriato test di sistema di una funzionalità del sistema sviluppato	Sì	Ogni membro ha eseguito il test di sistema di una funzionalità sviluppata.
Buona manutenibilità e integrabilità	Sì	<ul style="list-style-type: none">• Matrice di tracciabilità.• Uso di Design Pattern Façade• Documentazione del codice• Uso di architettura chiusa• Modello three-tier

6. Criteri di premialità

Criteri che, se rispettati, portano a punti aggiuntivi al progetto:

Criterio	Rispettato	Note
Uso adeguato di sistemi di build	Sì	Uso corretto di Maven.
Uso adeguato di un processo di continuous integration	Sì	Uso di SonarCloud. L'action di GitHub eseguiva: 1. Build dell'intero sistema 2. Check di warning, Code Smells, Security Issues, Duplication e Maintainability. 3. Esecuzione dei Test di Unità
Uso adeguato di tool di controllo della qualità	Sì	Uso di SonarCloud per code smells e bug report.
Adozione di processi di code review	Sì	Adoperato attraverso review del codice prima dell'accettazione delle pull request.
Uso adeguato di tool avanzati di testing	Sì	Uso di Mockito, Jacoco, SonarCloud e SeleniumIDE.