



# Università degli Studi di Salerno

Corso di Ingegneria del Software  
Classe 1 Resto 0  
Corso di Laurea in Informatica  
A.A. 2022/23

## Quiad System Design Document

Versione 2.0  
11/11/2022



**Partecipanti al progetto e scriventi**

Nome	Matricola
Di Pasquale Valerio	0512110638
Troisi Vito	0512109807

---

**Revision History**

Data	Versione	Descrizione	Autore
10/11/2022	1.0	Prima stesura SDD e Indice	D.P.V.
11/11/2022	2.0	Design goals	D.P.V. T.V.

## **Indice**

1. Introduzione
  - 1.1 Scopo del sistema
  - 1.2 Design goals
  - 1.3 Definizioni, acronimi, abbreviazioni
  - 1.4 Riferimenti
  - 1.5 Overview
2. Architettura del sistema proposto
  - 2.1 Overview
  - 2.2 Decomposizione in sottosistemi
  - 2.3 Mapping hardware/software
  - 2.4 Gestione dei dati persistenti
  - 2.5 Access control e sicurezza
  - 2.6 Global software control e concorrenza
  - 2.7 Boundary conditions
3. Servizi dei sottosistemi
4. Glossario

## Scopo del sistema

Il sistema **Quiad** si propone come **una piattaforma che consenta di esplorare la propria genealogia**, mediante un supporto grafico che consenta di creare e modificare il proprio albero genealogico. Agli antenati inseriti nell'albero, sarà possibile relare della documentazione storica, ricercata in un consistente archivio e reperibile secondo opportuni filtri.

Si anticipa, rispetto alle sezioni seguenti, che il sistema Quiad sarà concretizzato da una Web Application, progettata secondo l'architettura client-server. Gli utenti saranno dunque in grado di accedere alle funzionalità del sistema mediante un browser web e previa opportuna registrazione.

La fase di progettazione sarà guidata dagli omonimi obiettivi (design goals) di cui alla sezione immediatamente seguente.

## Design Goals

Il sistema Quiad è stato progettato considerando i seguenti design goals:

- **Criteri di affidabilità e sicurezza**

- I dati personali dell'utente (nome e cognome, data e luogo di nascita) dovranno essere soggetti ad un processo di cifratura prima di essere memorizzati nel sistema.
- Prima della memorizzazione della password di accesso degli utenti nel sistema, sarà generata a partire da essa una stringa hash generata mediante l'algoritmo *SHA-256* combinato con un algoritmo di *salt*.

- **Criteri di supportabilità**

- Il sistema deve prevedere l'inserimento di nuove tipologie di documentazione storica senza modifiche di funzionalità.

- **Criteri di performance**

- Il sistema deve essere progettato tenendo conto che la mole di dati che si intende memorizzare è significativamente elevata in termini di dimensioni in byte.
- Il sistema prevederà un meccanismo di caching lato server per le richieste legate alla *ricerca di*

*documentazione storica*. Il server manterrà in cache la risposta per la durata di un'ora.

- Il sistema prevederà un meccanismo di caching lato client per le richieste legate alla *ricerca di documentazione storica*. Il client manterrà in cache la risposta per la durata di due ore.

Si osservi che la tecnica di doppio caching proposta non consente all'utente di osservare "in tempo reale" eventuali modifiche ai documenti. Ciò costituisce un trade-off interessante tra funzionalità e performance, che è giudicato come significativamente a favore di queste ultime. Per il razionale completo, si consulti il Rationale Management Document (RMD), sezione 2.1.

Per una migliore comprensione dei design goals proposti e delle scelte effettuate, si rimanda al Documento di Analisi dei Requisiti (RAD), sezione 3.3 - Requisiti non funzionali, identificativi RNFR, RNFS, RNFP.

### **Definizioni, acronimi, abbreviazioni**

Una lista alfabetizzata di definizioni ed acronimi utili per la lettura della presente:

- RAD: Requirements Analysis Document.
- RMD: Rationale Management Document, i.e. il documento in cui è riportato il razionale dietro alcune scelte fatte.
- SDD: System Design Document, i.e. il presente documento.

Definizioni che non costituiscono acronimi, possono essere trovate nel Glossario, all'ultima sezione del presente documento.

### **Riferimenti**

Per garantire una migliore comprensione delle scelte effettuate in fase di System Design ed ivi descritte, si rimanda al Documento di Analisi dei Requisiti (RAD): particolare rilevanza è da associare ai requisiti non funzionali, trattati in fase di dichiarazione degli obiettivi di design; di particolare rilievo sono inoltre i modelli di sistema trattati nel RAD: scenari e casi d'uso, modello ad oggetti e modelli dinamici.

Inoltre, al fine di tracciare il razionale dietro le scelte effettuate, si consiglia la consultazione del Rationale Management Document nel quale saranno descritte le alternative proposte ed i criteri da cui esse sono state guidate.

## **Overview**

Il documento in questione è suddiviso in 4 sezioni principali:

1. La sezione introduttiva, che tratta i design goals adottati come linee guida della fase di progettazione e i principali trade-offs analizzati. Tale sezione si chiude con la presente overview.
2. La seconda sezione, l'architettura del sistema proposto, si aprirà con la decomposizione del sistema Quiad in sottosistemi. Saranno inoltre discusse diverse scelte di progettazione, dalla gestione dei dati persistenti, al controllo degli accessi.
3. La terza sezione descriverà poi i servizi messi a disposizione da ciascun sottosistema individuato nella sezione precedente, in termini delle interfacce che ciascuno espone.
4. Chiude il documento di System Design un glossario di termini particolarmente rilevanti nel contesto del dominio applicativo, delle soluzioni e nell'ambito del System Design Document stesso.

---

## **Architettura del sistema proposto**

Nella sezione seguente saranno dettagliate le specifiche di progettazione della piattaforma Quiad, che incarnano i design goals precedentemente espressi.

## **Overview**

A partire dalla decomposizione in sottosistemi e dalla specifica delle funzionalità di ognuno, saranno osservati: il mapping hardware/software dei sottosistemi, il trattamento

riservato ai dati persistenti e la matrice degli accessi degli utenti del sistema. Chiuderanno tale sezione, la descrizione del software control per Quiad ed eventuali boundary conditions.

L'architettura software che sarà dettagliata, di carattere client/server è *layered* nei suoi sottosistemi, proponendo la suddivisione di cui al paragrafo seguente.

### Controllo degli Accessi e Sicurezza

	Utente Non registrato	Utente Base	Utente Curatore	Utente Supervisore
<b>Home</b>	Visualizzare	Visualizzare	Visualizzare	Visualizzare
<b>Account</b>	Registrazione	Accesso	Accesso	Accesso
<b>Tree</b>		Visualizza Modifica	Visualizza Modifica	
<b>Document</b>		Visualizza	Visualizza Crea	Visualizza Modifica
<b>Promotion</b>		Visualizza Richiede		Visualizza Modifica

Gli **ospiti**, i.e. gli utenti non registrati al sistema *Quiad* possono navigare solo in una porzione limitata delle pagine. Nello specifico possono accedere esclusivamente ai servizi di Accesso/Registrazione utente o alla pagina principale della piattaforma.

Gli **utenti base** registrati a *Quiad*, ereditano le capacità degli *utenti ospite*, in aggiunta hanno il potere di costruire il proprio albero genealogico a partire da un nodo fulcro centrale, il quale rappresenta la loro persona. Inoltre possono ricercare la documentazione storica memorizzata in piattaforma e richiedere di essere promossi a *utente curatore*.

Gli **utenti curatori** di *Quiad*, oltre a essere comuni fruitori delle funzionalità classiche del sistema (come gli *utenti base*), hanno la capacità di caricare documentazione storica in piattaforma, che potrà essere successivamente visualizzata dagli utenti.

Gli **utenti supervisore** di Quiad rappresentano una figura di controllo del flusso di documentazione storica immagazzinata in piattaforma. Hanno il potere di respingere o approvare richieste di promozione a *utente curatore*, idem comparate per la documentazione storica.