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 4.12
31/12/2022

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.
16/11/2022	3.0	Servizi e operazioni	D.P.V.
18/11/2022	3.1	Introduzione a HW/SW Mapping e Global Software Control	D.P.V. T.V.
22/11/2022	4.0	Gestione dei dati persistenti	D.P.V. T.V.
22/11/2022	4.1	Gestione dei dati persistenti, ampliamenti	D.P.V. T.V.
22/11/2022	4.2	Correzione Design Goals	T.V.
23/11/2022	4.3	Global Software Control, concorrenza, assemblaggio	T.V.
28/11/2022	4.4	Modifica gestione dati persistenti (modello del database)	D.P.V. T.V.
29/11/2022	4.5	Aggiunta operazione a modulo TreeManager	T.V.
30/11/2022	4.6	Modifica gestione dati persistenti, modificato schema database	D.P.V. T.V.
02/12/2022	4.7	Eliminato schema non necessario	T.V.
08/12/2022	4.8	Eliminata sezione dati persistenti non necessaria, aggiunta versione DBMS utilizzato	T.V.
10/12/2022	4.9	Aggiunta sezione Concorrenza	D.P.V.
15/12/2022	4.10	Aggiunti riferimenti al mapping client side	D.P.V.
28/12/2022	4.11	Aggiunta specifica token in Sicurezza	T.V.
31/11/2022	4.12	Aggiunta annotazione al package diagram per sottosistemi	D.P.V. T.V.

Indice

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 mantenimento (e 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 un'ora.

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.

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

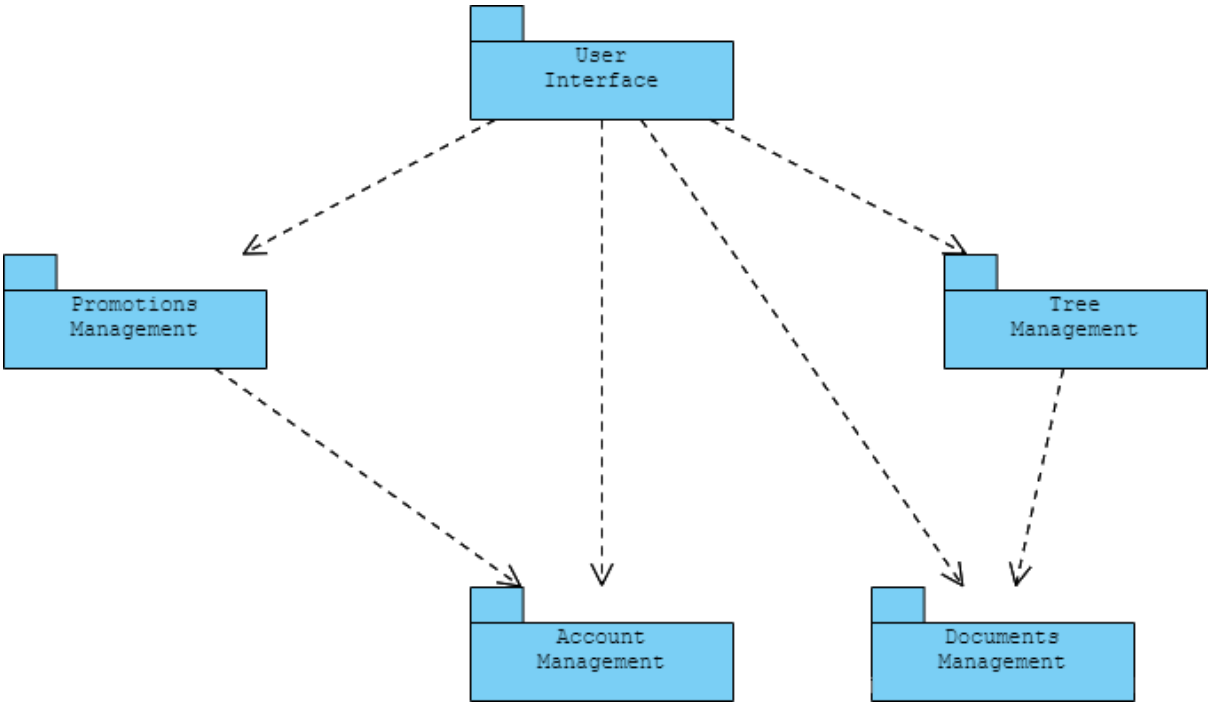
A partire dalla decomposizione in sottosistemi e dalla specifica delle funzionalità di ognuno, saranno osservati: il mapping hardware/software dei sottosistemi, il trattamento riservato ai dati persistenti e la matrice degli accessi degli utenti del sistema. Chiuderanno tale sezione, la descrizione del 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.

Decomposizione in sottosistemi

Di seguito la decomposizione del sistema, espressa mediante Package Diagram e Component Diagram. Le componenti descritte in quest'ultimo saranno riprese in seguito per illustrare la locazione delle medesime a runtime.

Package Diagram per i sottosistemi



Naturalmente il tutto poggia sullo strato di persistenza, discusso più avanti nel presente documento.

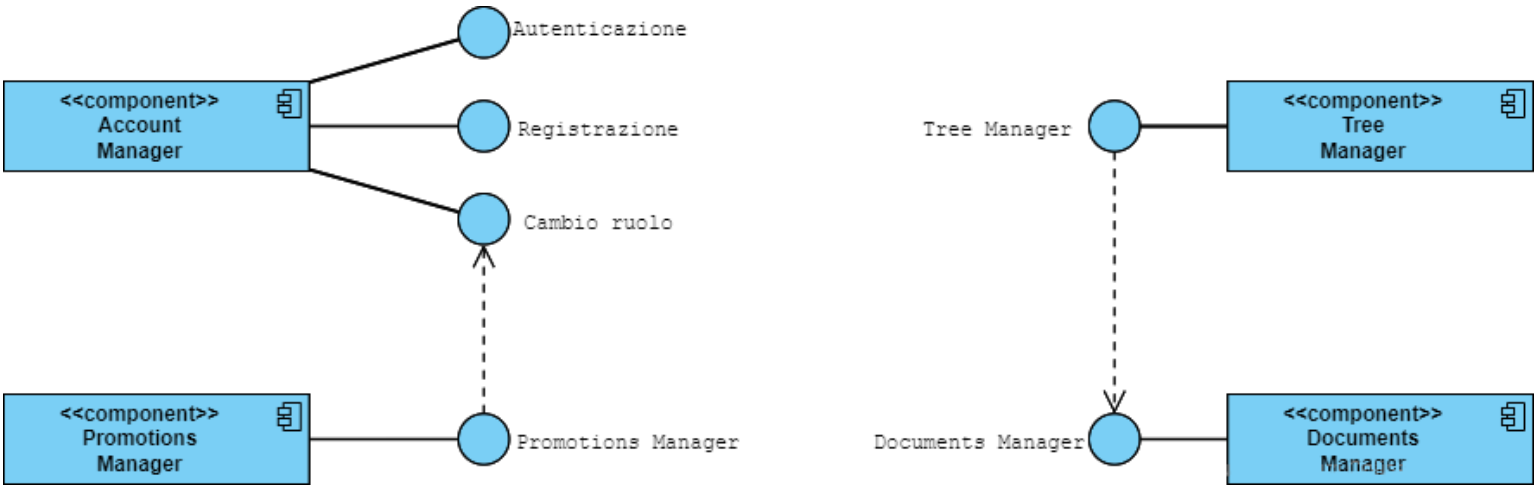
Mapping Hardware/Software

Il sistema è basato su un'architettura three tier. Nello specifico, il primo modulo server espone una REST API, la quale è interrogabile dal client, allo scopo di eseguire le operazioni previste dal sistema. Il secondo modulo server, espone le pagine HTML dell'applicazione, le quali forniscono ai Browser tutte le istruzioni su come effettuare le operazioni e/o visualizzare le informazioni. Il server che espone una REST API, scritto in JavaScript, interroga il Database MySQL per mezzo dell'ORM Prisma.

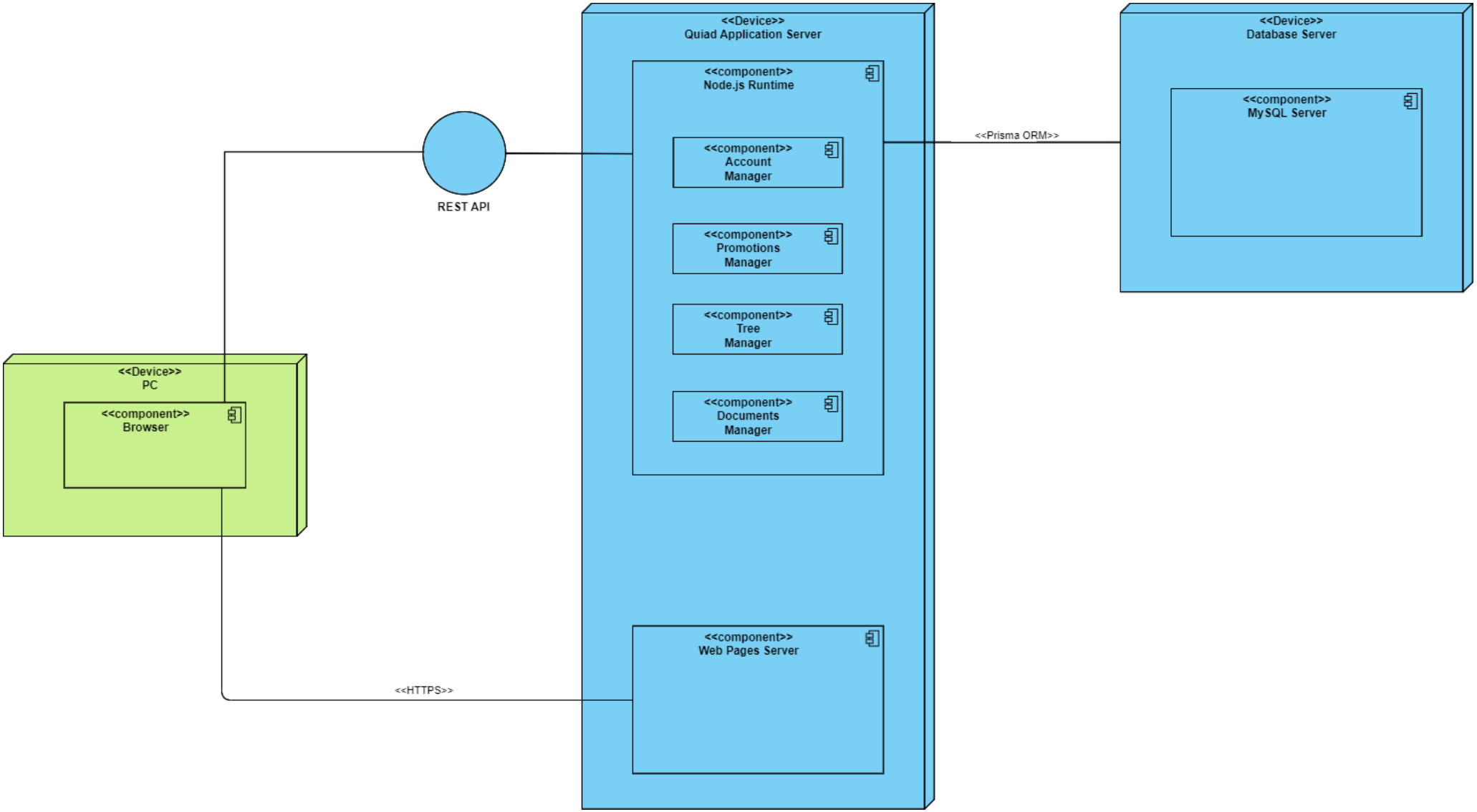
Si prevede di implementare il client mediante il paradigma single-page application, impiegando tecnologie quali Angular. Il server restituirà una suddivisione in moduli Angular delle pagine. La tecnica Ajax consente al client di richiedere i moduli in maniera asincrona. Per il razionale dietro tale scelta, si rimanda al RMD, sezione 2.3.

Seguono il component diagram ed il deployment diagram del sistema e dei sottosistemi come descritto fino al paragrafo presente.

Component Diagram



Deployment Diagram



Per un approfondimento sulla scelta legata alla REST API, e la predilezione della stessa rispetto al preprocessing, si consulti il RMD, sezione 2.2.

Gestione dei dati persistenti

Il sistema farà uso di un database relazionale gestito mediante MySQL 5.7 al fine di archiviare i seguenti dati:

- Gli alberi genealogici degli utenti, memorizzati mediante una struttura di nodi relazionati tra essi ricorsivamente.
- L'insieme dei ruoli utente associati a tutte le operazioni che questi possono svolgere.
- I documenti storici disponibili in piattaforma, associati al loro effettivo path di memorizzazione sul file system.
- Le categorie dei documenti storici di cui al punto precedente.
- La lista degli utenti registrati in piattaforma, associati al loro ruolo ed ai dati ad essi relativi.

Come anticipato ai punti precedenti, il sistema impiegherà direttamente il file system del server per memorizzare i documenti storici. Ciò consente di evitare di imporre sul database una mole eccessiva di dati, supponendo che i documenti, una volta approvati, difficilmente subiranno variazioni. Il formato dei documenti sarà PDF, con una dimensione compresa tra 1 Kb e 60 Mb, come indicato nella corrispondente sezione del RAD, paragrafo 3.5.2.

Inoltre, la necessità di operare con il file system inoltre, e gli eventuali ritardi che ciò potrebbe comportare, giustificano ulteriormente la necessità di un meccanismo di doppio caching.

Segue lo schema che descrive la base di dati, modellato in virtù della fase di Object Modeling (riferirsi, in tal senso, al RAD ed in particolar modo al Class Diagram al paragrafo 3.5.3) e dei punti precedentemente analizzati. La relazione tra i ruoli e le operazioni diventerà più chiara consultando la sezione "Controllo degli Accessi e Sicurezza" del presente documento.

Access Control e Sicurezza

	Utente Non registrato	**Utente** Base	**Utente** Curatore	**Utente** Supervisore
Account	Registrazione	Login Logout	Login Logout	Login Logout Cambio ruolo
Tree	-	Visualizza Inserimento nodo Modifica nodo Lega documento Slega documento Elimina nodo	Visualizza Inserimento nodo Modifica nodo Lega documento Slega documento Elimina nodo	-
Document	-	Visualizza Ricerca	Visualizza Ricerca Inserimento	Visualizza Ricerca Approvazione
Promotion	-	Visualizza Richiesta	-	Visualizza Approvazione

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.

Il client memorizzerà un token di autenticazione, in particolare un JSON Web Token, firmato con una chiave privata

generata lato server, cosicché solo quest'ultimo possa scrivervi e verificarne la validità. Ogni richiesta autenticata dovrà contenere il token suddetto, secondo lo schema Bearer.

Global Software Control e concorrenza

Come osservato ai paragrafi precedenti, il sistema sarà implementato come una Web Application, seguendo l'architettura client/server. Il server si occuperà di gestire le richieste del client ed assolvere alle stesse, interrogando la base di dati e producendo una risposta (e.g. una pagina Dynamic HTML).

Si osservi infine che il framework Prisma ORM consente la gestione automatizzata della concorrenza per quanto concerne le transazioni sul database. L'architettura di Node.js è nativamente ad eventi, ed è asincrono per natura, non prevedendo richieste bloccanti. La richiesta di un client viene automaticamente mappata sui thread interni, consentendo al server di continuare l'accettazione di richieste ulteriori.

Boundary Conditions

Il sistema non prevede boundary conditions di nota.

Servizi dei sottosistemi e operazioni

Le tavole di cui alle pagine seguenti riassumono i servizi di ciascun modulo (sottosistema). Nel caso in cui il modulo coincida con l'unico sottoservizio che offre, sarà riportata un'intestazione del tipo "Modulo/Servizio:". In tal caso saranno riportate le operazioni del servizio corrispondente.

Modulo: Account Management	
Autenticazione	Consente ad un utente registrato di autenticarsi presso il sistema o disconnettersi da esso.
Registrazione	Consente ad un utente non registrato di registrarsi presso il sistema.
Cambio ruolo	Consente ad un utente supervisore di modificare il ruolo (e.g. per una promozione) di un utente registrato.

Servizio: Autenticazione	
Login	Consente ad un utente registrato di autenticarsi presso il sistema.
Logout	Consente di terminare la conversazione con il sistema.

Modulo/Servizio: Tree Management	
Visualizza	Consente di visualizzare la struttura del proprio albero genealogico.
Inserimento nodo	Consente di aggiungere un nodo al proprio albero genealogico.
Modifica nodo	Consente di modificare i dati un nodo esistente.
Lega documento	Consente di aggiungere un riferimento ad un documento ad un nodo.
Slega documento	Consente di eliminare un riferimento ad un documento da un nodo
Elimina nodo	Consente di eliminare un nodo esistente diverso dal proprio.

Modulo/Servizio: Document Management	
Visualizza	Consente di visualizzare un dato documento.
Ricerca	Consente ad un utente base o supervisore di ricercare documenti applicando opportuni filtri di ricerca.
Inserimento	Consente ad un utente curatore di effettuare una nuova richiesta di inserimento.
Approvazione	Consente ad un utente supervisore di approvare o rigettare un documento inserito.

Modulo/Servizio: Promotions Management	
Visualizza	Consente di visualizzare una richiesta di promozione.
Richiesta	Consente ad un utente base di richiedere la propria promozione a utente curatore.
Approvazione	Consente ad un utente supervisore di approvare o rigettare una richiesta di promozione.