



SDD

System Design Document

SearchQS

Riferimento	SDD_ver.1.0
Versione	1.0
Data	11/04/2024
Destinatario	Prof. Fabio Palomba
Presentato da	Gianluca Scisciolo
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
02/11/2023	0.1	Prima stesura	GS
02/11/2023	0.2	Scrittura delle sezioni 1 e 2	GS
06/11/2023	0.3	Scrittura sezioni 3.1 e 3.2	GS
07/11/2023	0.4	Scrittura sezioni 3.3 e 3.4	GS
01/12/2023	0.5	Scrittura sezioni 3.5, 3.6, 3.7 e 3.8	GS
04/12/2023	0.6	Scrittura sezioni 4 e 5	GS
04/12/2023	0.7	Revisione documento	GS
15/12/2023	0.8	Revisione documento	GS
25/01/2024	0.9	Revisione documento	GS
06/03/2024	0.10	Revisione documento	GS
11/04/2024	1.0	Revisione finale documento	GS

Team members

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
Gianluca Scisciolo	Software Engineer	GS	g.scisciolo@studenti.unisa.it

Sommario

1.	Introduzione.....	4
1.1.	Scopo del sistema.....	4
1.2.	Obiettivi di design.....	4
1.3.	Definizioni, acronimi e abbreviazioni.....	7
1.4.	Riferimenti.....	7
1.5.	Organizzazione del documento.....	8
2.	Architettura del sistema corrente.....	9
3.	Architettura del sistema proposto.....	10
3.1.	Panoramica sulla sezione.....	10
3.2.	Decomposizione in sottosistemi.....	10
3.2.1.	Component Diagram UML del sistema proposto.....	11
3.2.2.	Vista dettagliata di ogni sottosistema.....	11
3.2.3.	Diagramma architetturale.....	12
3.2.4.	Sottosistemi.....	13
3.3.	Mapping hardware / software.....	16
3.4.	Gestione dei dati persistenti.....	17
3.5.	Progettazione del database:.....	17
3.6.	Controllo degli accessi e sicurezza:.....	20
3.7.	Controllo globale del software.....	20
3.8.	Condizioni limite.....	21
3.8.1.	Avvio del sistema.....	21
3.8.2.	Spegnimento del sistema.....	22
3.8.3.	Fallimento del sistema.....	23
3.8.4.	Errore di accesso ai dati persistenti.....	24
4.	Servizi dei sottosistemi.....	25
4.1.	Sottosistema authentication.....	25
4.2.	Sottosistema user area.....	25
4.3.	Sottosistema analysis.....	26
5	Glossario.....	27

1. Introduzione

1.1. Scopo del sistema

Il sistema che si vuole realizzare ha come obiettivo l'analisi di un sistema quantistico per poter individuare i quantum code smells presenti, ovvero, i seguenti 8:

- Non-parameterized Circuit (NC).
- no-alignment between the Logical and Physical Qubits (LPQ).
- Idle Qubits (IdQ).
- Initialization of Qubits (IQ).
- Intermediate Measurement (IM).
- Long Circuit (LC).
- Repeated set of Operations on Circuit (ROC).
- use of Customized Gates (CG).

L'utente può eseguire delle analisi (ogni analisi è composta da analisi statica e analisi dinamica) sui vari sistemi quantistici. Una analisi coinvolge solo un tipo di transpilazione oppure nessuna transpilazione quindi, se per esempio sceglie 3 transpilazioni e la modalità senza transpilazione allora vengono eseguite in totale 4 analisi ognuna delle quali è costituita da analisi statica più analisi dinamica. Le transpilazioni offerte dal sistema sono le seguenti:

- original.
- ibm_perth.
- ibm_sherbrooke.
- rpcx.
- simple.

1.2. Obiettivi di design

In questa sezione sono presenti i design goals che rappresentano le qualità che il sistema vuole ottenere.

Ogni decisione di design sarà presa considerando i design goals presenti in questa sezione.

I design goals saranno suddivisi nelle seguenti categorie:

- Performance: determinano i requisiti di spazio e velocità imposti sul sistema.
- Dependability: determinano lo sforzo che deve essere speso per minimizzare i fallimenti del sistema e le loro conseguenze.
- Maintenance: determinano lo sforzo necessario per modificare e mantenere il sistema dopo il suo rilascio.
- End User: determinano le qualità che si vogliono raggiungere dal punto di vista dell'utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ogni design goal è descritto da:

- Rank: specifica un valore di priorità compreso tra 1 (massimo) e 10 (minimo).
- ID Design Goal: un identificatore univoco e un nome esplicativo.
- Descrizione: descrizione del design goal.
- Categoria: categoria di appartenenza del design goal.
- RNF di origine: il Requisito Non Funzionale che lo ha generato.

Design Goals

Rank	ID Design Goal	Descrizione	Categoria	RNF di origine
2	DG_1 Tempi di risposta	Il sistema dovrà fornire la risposta alle operazioni degli utenti in tempi brevi: circa 5 secondi per ogni operazione tranne per le operazioni dell'analisi che ruberà più tempo, circa massimo 10 secondi.	Performance	RNF_Pr_2.
3	DG_2 Creazione database relazionale gestito con MySQL	Creazione di un sistema che utilizza un database relazionale gestito tramite il RDBMS MySQL.	Performance	RNF_In_1.
3	DG_3 Affidabilità delle operazioni	Il sistema deve garantire che tutte le operazioni avvengano con successo.	Dependability	RNF_Af_1.
9	DG_4 Fallimento di sistema	Il sistema deve gestire le situazioni in cui fallisce inviando delle notifiche all'utente tramite messaggi di errore.	Dependability	RNF_Af_3.
9	DG_5 Disponibilità del sistema	Il Sistema deve garantire la massima disponibilità, con un limite di 48 ore all'anno di downtime.	Dependability	RNF_Pr_1.
1	DG_6 Sicurezza dei dati	Il sistema deve garantire la massima sicurezza dei dati tramite l'uso di protocolli di comunicazione sicuri, conservando in maniera sicura i	Dependability	RNF_Af_2, RNF_Le_1,

		dati persistenti e assicurando la visualizzazione dei dati solo agli utenti che hanno diritto ad accedervi.		
5	DG_7 Manutenibilità	Il sistema dovrà essere semplice da mantenere.	Maintenance	RNF_Su_1.
7	DG_8 Estendibilità	Il sistema dovrà essere semplice da estendere e dovrebbe essere semplice inserire nuove funzionalità.	Maintenance	RNF_Su_2.
4	DG_9 Implementazione di una web application testabile con l'hardware disponibile	Si deve implementare una web application utilizzando l'hardware disponibile e garantire una branch coverage del 70 – 80%.	Maintenance	RNF_Im_1, RNF_Im_2, RNF_Im_3.
4	DG_10 Facilità d'uso	Il sistema deve essere semplice da comprendere ed utilizzare e, inoltre, il sistema deve permettere un utilizzo del sistema chiaro e di facile comprensione anche senza la lettura di documenti.	End User	RNF_Us_1, RNF_Us_3.
7	DG_11 Interfaccia intuitiva	L'interfaccia del sistema deve essere tale da permettere l'esecuzione delle azioni in maniera chiara ed intuitiva.	End User	RNF_Us_2.
		Il sistema deve essere facilmente deployabile,		

8	DG_12 Deployment semplificato	esponendo la web application sulla rete e accedendo a tutti i servizi esterni necessari.	End User	RNF_Pa_1.
10	DG_13 Interfaccia con le utility di sistema per l'admin	Il Sistema deve interagire correttamente con le utility del sistema operativo sul quale viene eseguito il server.	End User	RNF_Op_1.

1.3. Definizioni, acronimi e abbreviazioni

In questa sezione sono presenti alcune definizioni presenti in questo documento.

- Sottosistema: un sottoinsieme di servizi del dominio applicativo formato da servizi legati da una relazione funzionale.
- Design Goal: le qualità sulle quali il sistema deve essere focalizzato.
- Dati Persistenti: dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque vengono salvati.
- Mapping Hardware/Software: studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.

1.4. Riferimenti

di seguito sono riportati altri documenti utili durante la lettura.

- [Statement Of Work \(SOW\).](#)
- [Business Case \(BC\).](#)
- [Requirements Analysis Document \(RAD\).](#)
- [DataBase Design Document \(DBDD\).](#)
- [Object Design Document \(ODD\).](#)
- [Test Plan \(TP\).](#)
- [Test Case Specification \(TCS\).](#)
- [Test Incident Report \(TIR\).](#)
- [Test Incident Report Table \(TIRT\).](#)
- [Test Summary Report \(TSR\).](#)
- [Manuale Di Installazione \(MDI\).](#)
- [Manuale Utente \(MU\).](#)

1.5. Organizzazione del documento

Questo documento è costituito da 4 sezioni:

- Introduzione: descrizione dello scopo del sistema e degli obiettivi di design che il sistema vuole raggiungere.
- Architettura software corrente: descrizione dello stato attuale dell'architettura del software già presente.
- Architettura software proposta: descrizione dei seguenti punti:
 - come il sistema sarà definito.
 - come il sistema sarà partizionato in sottosistemi.
 - il mapping Hardware / Software.
 - gestione dei dati persistenti.
 - presentazione della struttura dei singoli sottosistemi e le boundary conditions riguardanti l'intero sistema.
- Glossario: contiene la lista dei termini usati nel documento con la loro descrizione.

2. Architettura del sistema corrente

Il sistema QSmell ha solamente un package chiamato qsmell in cui sono definiti all'interno i seguenti packages:

- smell: contiene le classi in cui sono definiti i quantum code smells da analizzare.
- utils: contiene dei metodi utili per le matrici utilizzate per l'analisi dinamica e per le classi dei quantum code smells in generale.

3. Architettura del sistema proposto

3.1. Panoramica sulla sezione

Il sistema che si vuole realizzare si baserà sullo stile architetturale Three Tier, implementato utilizzando il design pattern architetturale MVT (Model View Template) con l'aggiunta di un livello Service per semplificare il compito del livello View e di una cartella Static in cui sono presenti tutti i file statici come le immagini, i fogli di stile e i file JavaScript utilizzato dal livello Template. Verrà quindi chiamato: MVTs (Model View Template Service).

È stato scelto questo stile poiché è indicato per lo sviluppo di web applications perché è in grado di separare la logica di presentazione dalla logica di elaborazione migliorando varie qualità, tra cui :

- La leggibilità del codice.
- La manutenzione del codice.
- Il riuso del codice.

Anche il design pattern architetturale MVTs è in grado di ottenere i 3 punti sopra riportati e inoltre è la scelta più utilizzata quando si lavora con il micro-framework Flask che si utilizzerà nel sistema che verrà realizzato.

Cosa verrà utilizzato:

- verrà utilizzato l'IDE Visual Studio Code v.1.87.2;
- per il lato front-end verrà utilizzato:
 - HTML5.
 - CSS3.
 - Bootstrap v.5.0.
 - JavaScript.
 - Template Jinja2 v.3.1.2 per inserire del codice Python nei file HTML5.
- Per il lato back-end verrà utilizzato:
 - Python v.3.11.5 con interprete conda.
 - L'SDK Qiskit v.0.45.1 per il lato riguardante la computazione quantistica.
 - Il micro framework Flask v.3.0.0
 - per la gestione del database verrà utilizzato:
 - il RDBMS MySQL v.8.0.19 che utilizzerà il linguaggio SQL.

3.2. Decomposizione in sottosistemi

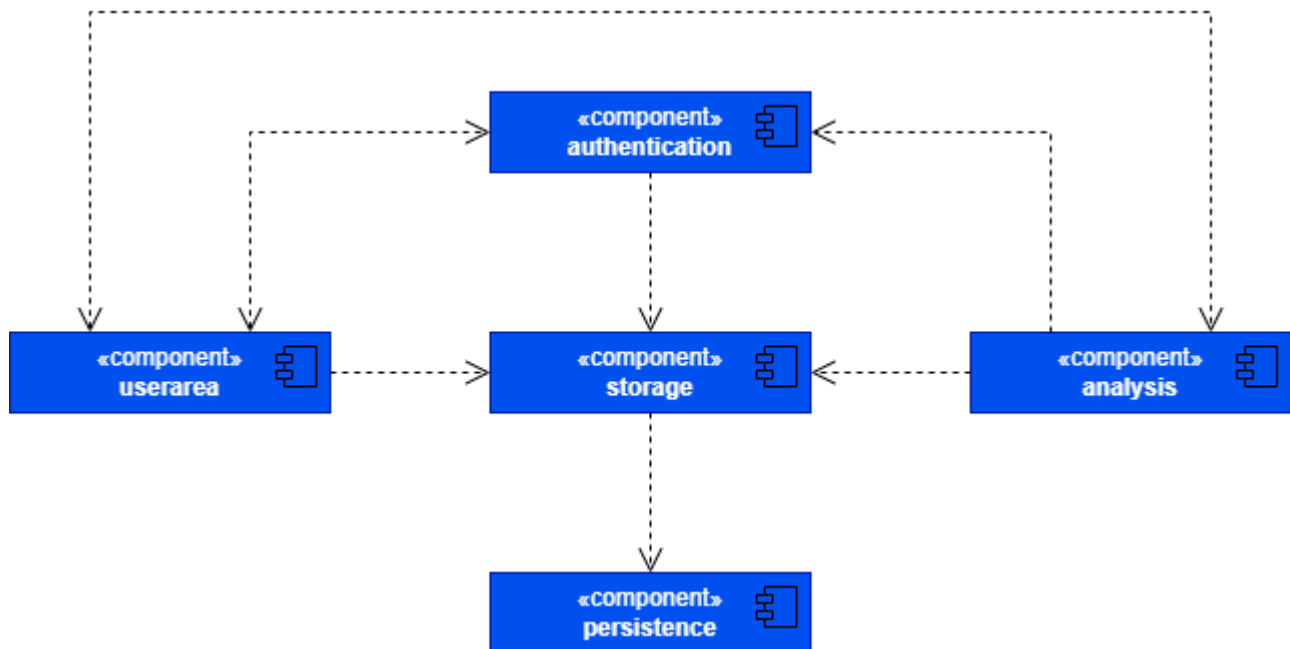
I sottosistemi individuati sono:

- Authentication
- User area
- Analysis
- Persistence: gestisce la persistenza dei dati con un database relazionale.
- Storage: si interpone tra i vari sottosistemi e il sottosistema di Persistenza.

Il sottosistema persistence utilizzerà come componente COTS il RDBMS MySQL.

3.2.1. Component Diagram UML del sistema proposto

Nella seguente sezione sono mostrate le dipendenze tra i vari sottosistemi attraverso un component diagram UML



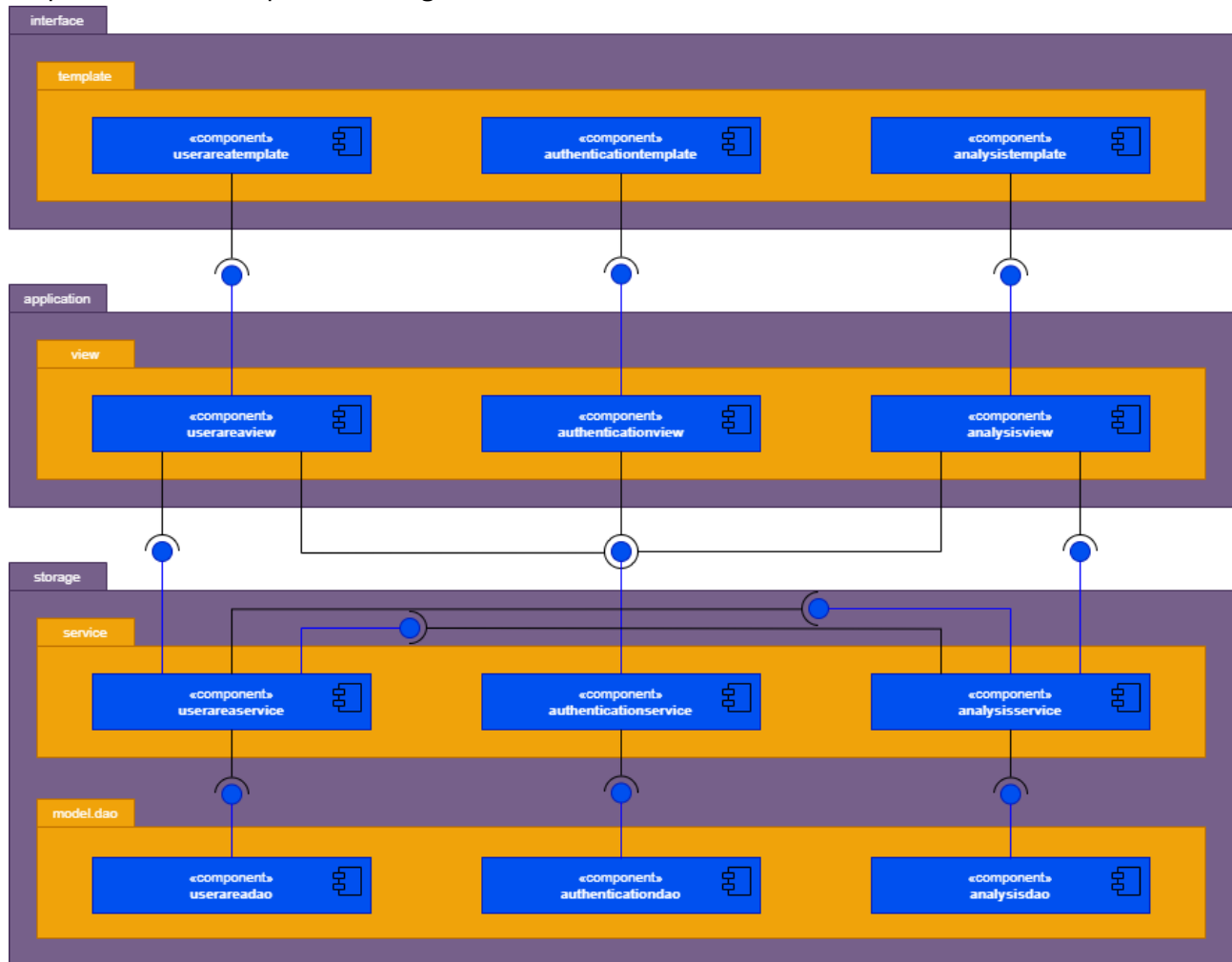
3.2.2. Vista dettagliata di ogni sottosistema

In questa sezione è riportata una vista dettagliata di ogni sottosistema:

- Template: contiene i vari file HTML che l'utente visualizzerà. Quindi, si occuperà della gestione dell'interfaccia utente del sottosistema in questione.
- View: si occupa della gestione delle richieste HTTP del sottosistema in questione. Interagisce con il template e con il service.
- Service: si occupa della logica di business. Interagisce con il DAO (presente nel Model) e ritorna una risposta alla view.
- DAO: si occupa di fornire l'accesso ai dati persistenti. È situato nel Model e ritorna una risposta al service.

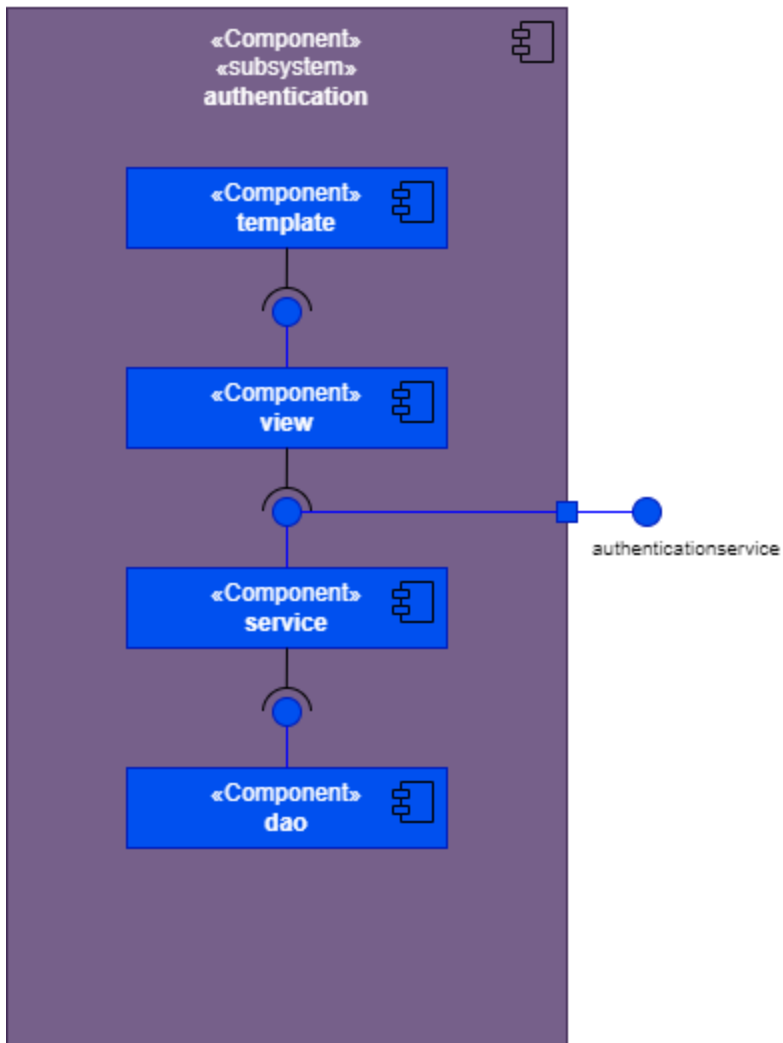
3.2.3. Diagramma architetturale

In questa sezione è riportato il diagramma architetturale dell'intero sistema.

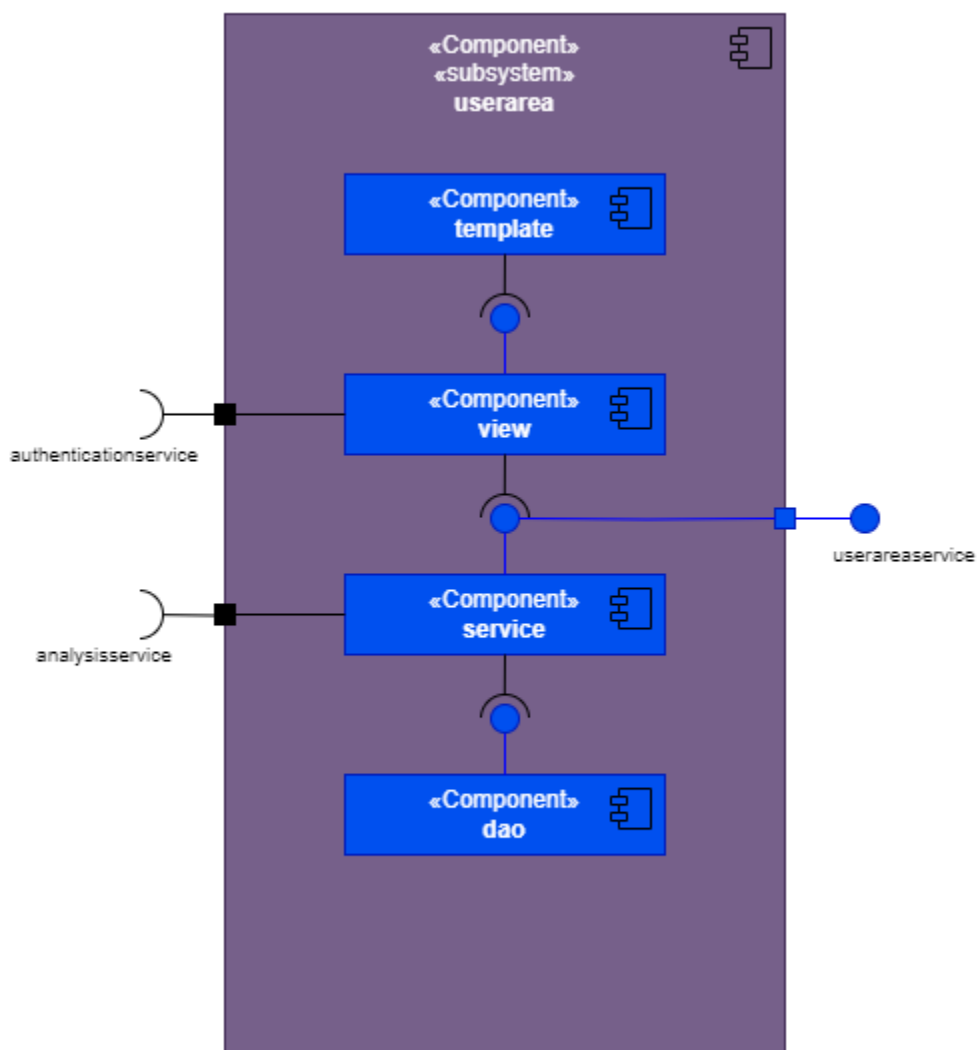


3.2.4. Sottosistemi

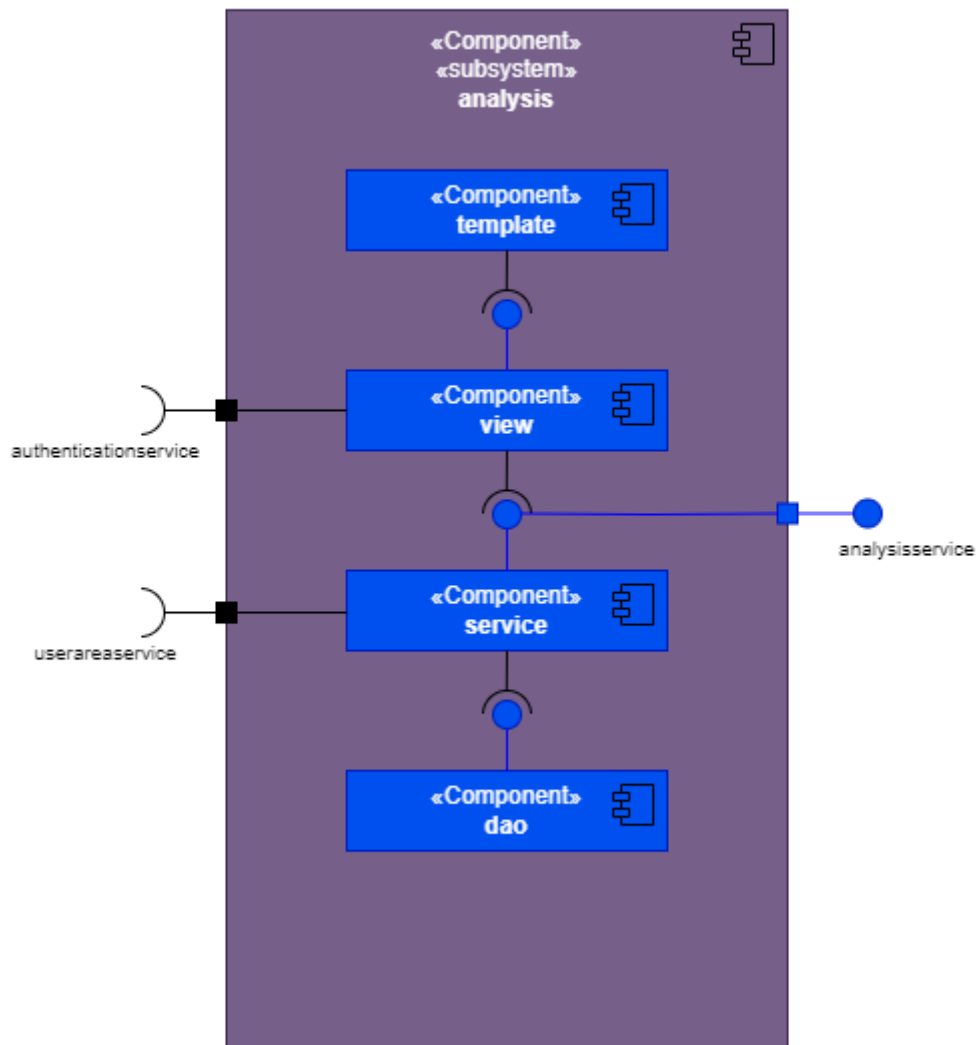
In questa sezione sono riportati i vari sottosistemi tramite dei component diagrams.
Sottosistema Authentication:



Sottosistema User area:



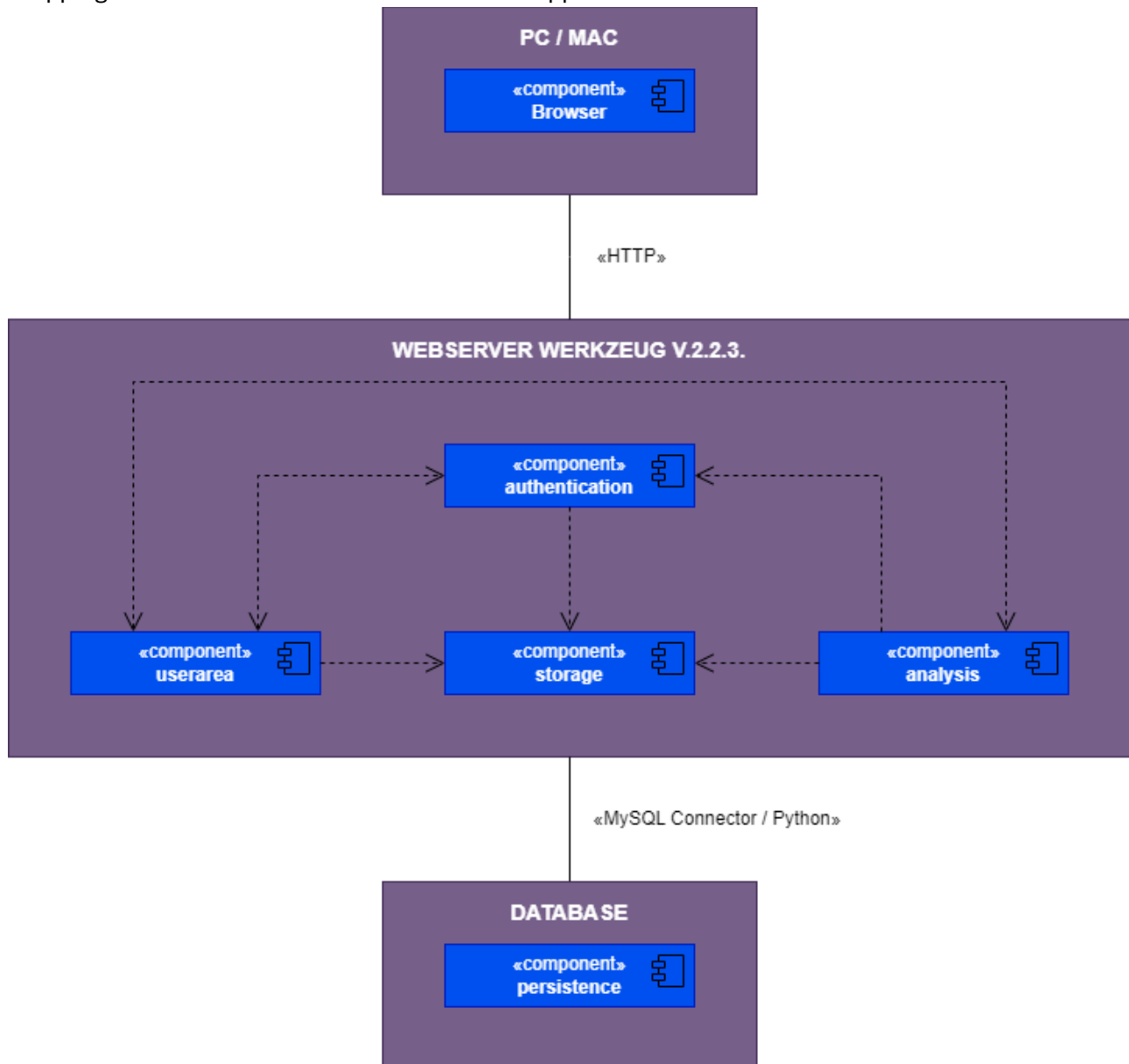
Sottosistema Analysis:



3.3. Mapping hardware / software

La web application che si vuole realizzare si basa su una piattaforma hardware costituita da un server che risponde alle richieste effettuate dai clienti da una qualsiasi macchina con un browser ed una connessione ad Internet. Il nostro sistema è una web application che è situata su un web server e, siccome non è un sistema distribuito allora è situato su un solo nodo.

In questa sezione è riportato un Deployment Diagram UML che descrive il mapping hardware / software della nostra web application.



3.4. Gestione dei dati persistenti

Per gestire il salvataggio dei dati persistenti del sistema si utilizzerà un database relazionale gestito tramite l'utilizzo del RDBMS MySQL in locale su un'unica macchina.

È stato scelto un RDBMS poiché possiamo contare sui seguenti aspetti:

- **Privatezza dei dati:** un RDBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
- **Affidabilità dei dati:** un RDBMS offre dei metodi per salvare copie dei dati e per ripristinare lo stato della base di dati in caso di guasti hardware e software.
- **Atomicità delle operazioni:** un RDBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base. L'atomicità delle transazioni permette di mantenere uno stato della base di dati consistente con la realtà modellata.

3.5. Progettazione del database:

Il database verrà chiamato: searchqs.

Per motivi di leggibilità del documento si è deciso di creare un documento apposito per la sua progettazione prendendo come input lo schema E-E-R del database individuato a partire dal Class Diagram Entity presente nel RAD ed ottenendo come output lo schema logico relazionale finale.

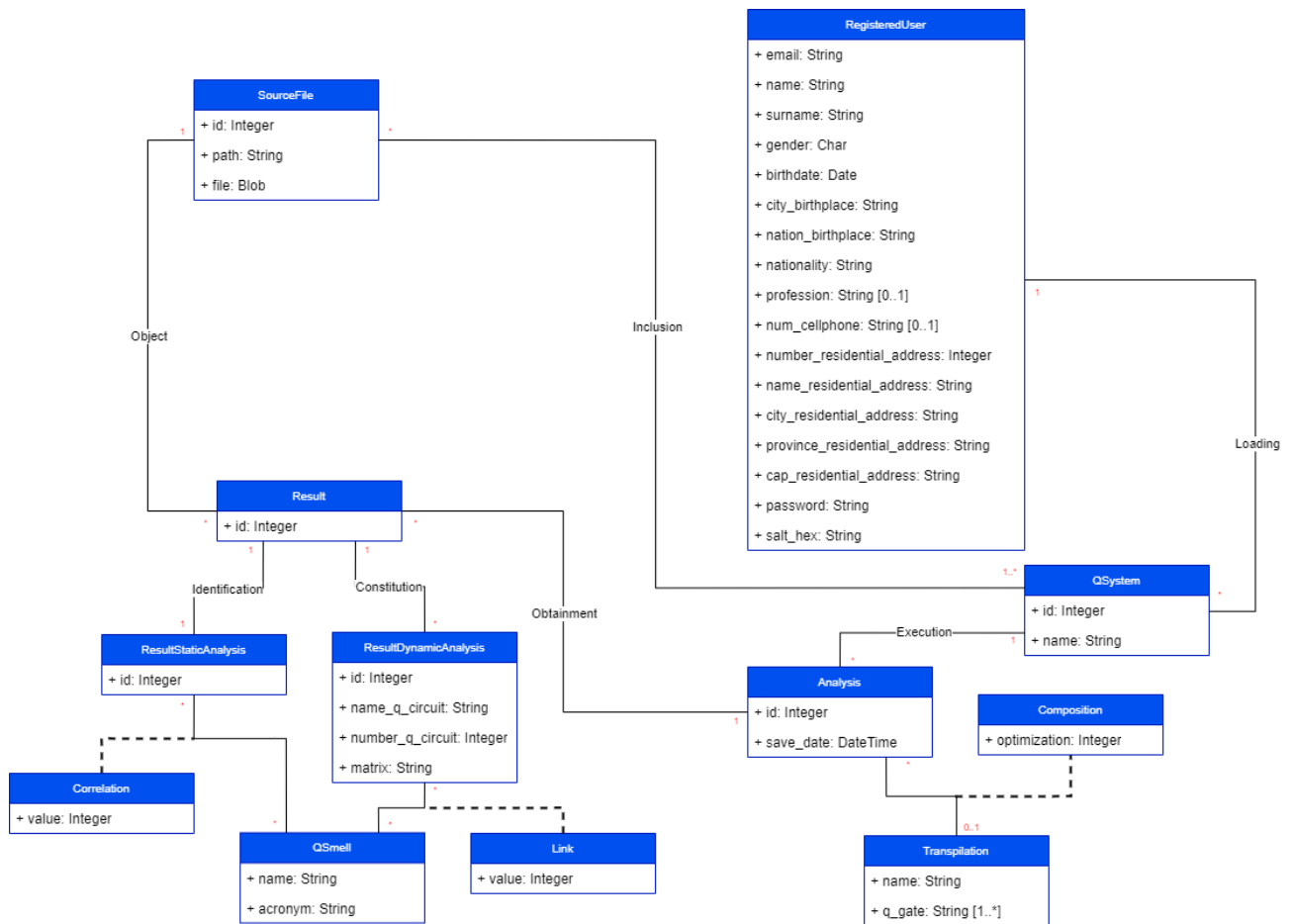
Il documento è stato allegato nella sezione 1.4 di questo documento con il seguente nome:

- DataBase Design Document (DBDD).

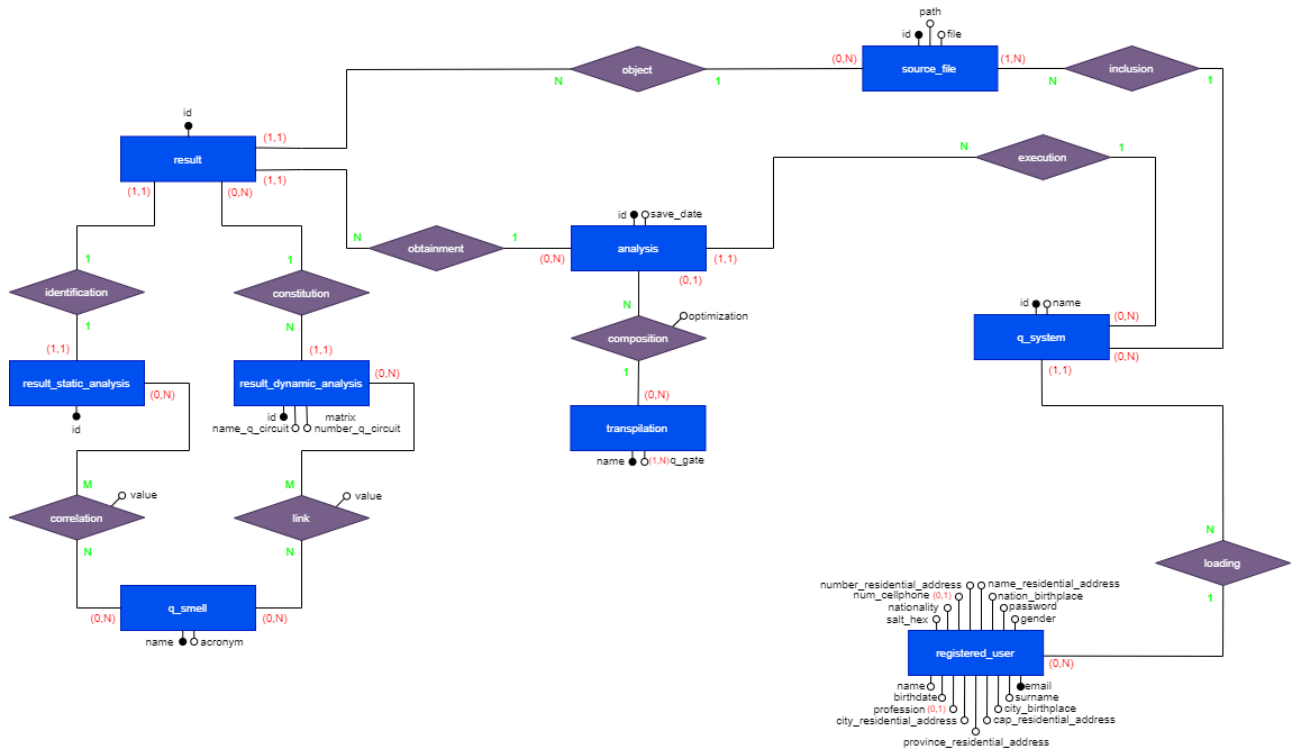
In questo documento verrà riportato solamente:

- Il Class Diagram Entity iniziale.
- Lo schema E-E-R iniziale.
- Lo schema E-R finale ottenuto.
- Lo schema logico relazionale finale ottenuto.

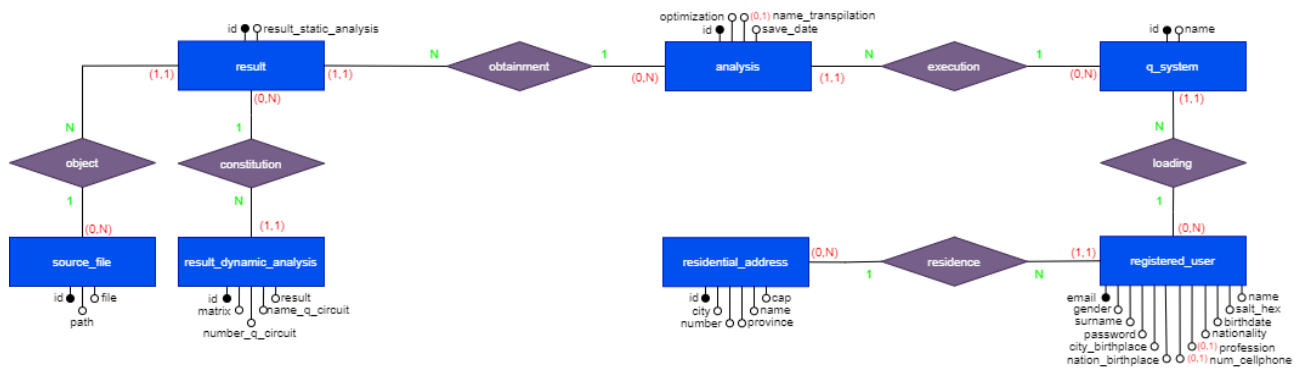
Class Diagram Entity iniziale:



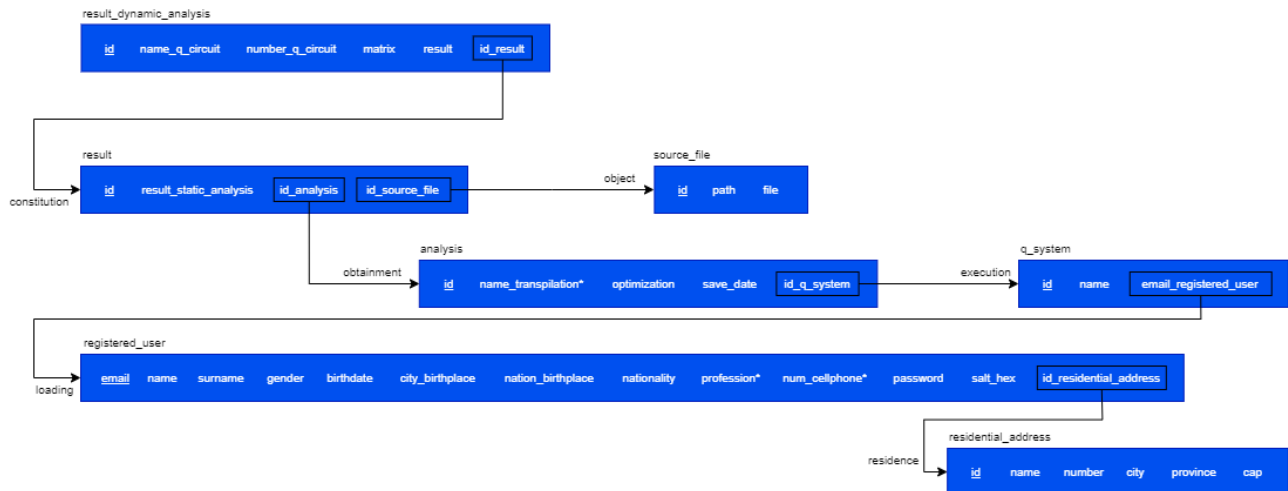
Schema E-E-R iniziale:



Schema E-R finale:



Schema logico relazionale finale:



3.6. Controllo degli accessi e sicurezza:

In questa sezione è riportata la matrice degli accessi per tenere traccia degli attori del sistema e a quali servizi ogni attore può accedere:

Oggetti	Attori	
	Guest	Registered user
authentication	display_form_registration	logout
	registration	
	display_form_login	
	login	
user_area		display_user_area
		display_form_deletion_account
		deletion_account
		display_personal_data
		modification_personal_data
analysis		display_analysis_area
		display_form_loading_q_system
		loading_q_system
		execution_analyses
		display_names_transpilation
		display_analyses_transpilation_selected
		display_analysis
		display_form_deletion_analysis
		deletion_analysis

3.7. Controllo globale del software

Il sistema che si vuole realizzare sarà una web application la cui sua funzionalità avviene tramite dei comandi inviati dall'utente tramite l'utilizzo di una interfaccia grafica tramite cui l'utente compilerà dei form e successivamente premere un pulsante oppure, nel caso in cui non sono presenti dei form, premere direttamente un pulsante. Quest'ultimo invierà una richiesta HTTP di tipo POST oppure GET alla view. La view gestirà la richiesta HTTP e richiamerà il metodo service interessato e quest'ultimo seguirà uno delle seguenti azioni:

- gestirà l'operazione da eseguire e restituirà un output alla view.
- richiamerà 1 o più DAO che interrogheranno il database restituendo un risultato finale.
- richiamerà un altro metodo service.

Successivamente la view ritornerà un output al template interessato e quest'ultimo visualizzerà una pagina HTML utilizzando anche possibili fogli di stile CSS e / o possibili file JavaScript e / o possibili immagini presenti nella cartella static del sistema.

3.8. Condizioni limite

In questa sezione sono presenti le boundary condition che riguardano:

- l'avvio del sistema.
- Lo spegnimento del sistema.
- Il fallimento del sistema.
- L'errore di accesso ai dati persistenti.

3.8.1. Avvio del sistema

Identificativo		UCBC_1 – Avvio del sistema	Data	
			Versione	
			Autori	Scisciolo Gianluca
Descrizione		Lo UC permette il riavvio del sistema.		
Attore principale		Admin		
Attori secondari		NA		
Entry condition		L'admin accede al server.		
Exit condition (on success)		Il sistema viene avviato correttamente.		
Exit condition (on failure)		Il sistema non viene avviato.		
Flusso di eventi principale				
1	Admin	Esegue sulla macchina il comando per avviare il sistema.		
2	Sistema	Verifica se i dati persistenti siano sani e se la risposta è affermativa allora rende disponibili i suoi servizi e rende le sue funzionalità disponibili agli utenti.		
I Flusso di eventi alternativo: i dati persistenti sono danneggiati				
2.a1	Sistema	Notifica all'admin che i dati persistenti sono danneggiati e non effettua l'avvio.		
2.a2	Admin	Corregge i dati persistenti.		
2.a3	Sistema	Il sistema effettua le modifiche.		
2.a4	Admin	Esegue il passaggio 1.		

3.8.2. Spegnimento del sistema

Identificativo	UCBC_2 – Spegnimento del sistema	Data	
		Versione	
		Autori	Scisciolo Gianluca
Descrizione	Lo UC permette lo spegnimento del sistema.		
Attore principale	Admin		
Attori secondari	NA		
Entry condition	L'admin accede al server AND Il sistema è stato precedentemente avviato AND Il sistema non è stato ancora spento.		
Exit condition (on success)	Il sistema viene spento correttamente.		
Exit condition (on failure)	Il sistema non viene spento.		
Flusso di eventi principale			
1	Admin	Invia un segnale di spegnimento al sistema.	
2	Sistema	Verifica che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.	
I Flusso di eventi alternativo: Ci sono connessioni ancora aperte			
2.a1	Sistema	Notifica all'admin che ci sono ancora connessioni aperte verso l'esterno.	
2.a2	Sistema	Attende una quantità di tempo per rispondere a eventuali richieste dall'esterno, non generando nuove connessioni se non allo scopo di rispondere a richieste già in corso.	
2.a3	Sistema	Controlla che non ci siano connessioni ancora aperte da o verso l'esterno e, se non ci sono, termina l'esecuzione del sistema.	
2.a4	Sistema	Notifica l'admin dell'avvenuto spegnimento del sistema.	
II Flusso di eventi alternativo: Ci sono connessioni ancora aperte			
2.a3.a1	Sistema	interrompe le connessioni verso l'esterno.	
2.a3.a2	Sistema	Notifica l'Amministratore dell'avvenuto spegnimento del sistema e del numero di connessioni interrotte.	

3.8.3 Fallimento del sistema

Identificativo	UCBC_3 – Fallimento del sistema	Data	
		Versione	
		Autori	Scisciolo Gianluca
Descrizione	Lo UC definisce il comportamento del Sistema in caso di fallimento.		
Attore principale	Admin		
Attori secondari	NA		
Entry condition	Il Sistema viene terminato inaspettatamente.		
Exit condition (on success)	Il Sistema viene riavviato correttamene.		
Exit condition (on failure)	Il Sistema non viene riavviato.		
Flusso di eventi principale			
1	Admin	Include UCBC_1	

3.8.4. Errore di accesso ai dati persistenti

Identificativo	UCBC_4 – Errore di accesso ai dati persistenti	Data	
		Versione	
		Autori	Scisciolo Gianluca
Descrizione	L'UC descrive il comportamento del sistema qualora fosse impossibile accedere ai dati persistenti o questi risultassero corrotti.		
Attore principale	Admin		
Attori secondari	NA		
Entry condition	Il Sistema non può accedere ai dati persistenti OR I dati persistenti risultano corrotti.		
Exit condition (on success)	Il Sistema riprende il normale funzionamento.		
Exit condition (on failure)	Il Sistema non riprende il normale funzionamento.		
Flusso di eventi principale			
1	Sistema	Notifica l'amministratore dell'impossibilità di accedere ai dati persistenti.	
2	Sistema	Cessa di processare eventuali richieste dall'esterno e risponde a tutte le richieste con un messaggio di errore.	
3	Admin	Include UCBC_2	
4	Admin	Ripristina l'accessibilità o la sanità dei dati persistenti.	
5	Admin	Include UCBC_1	

4. Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

4.1. Sottosistema authentication

Servizio	Descrizione	Interfaccia
Display form registration	Permette la visualizzazione del form di registrazione.	i_authentication_service.IAuthenticationService
Registration	Permette la registrazione alla web application SearchQS.	i_authentication_service.IAuthenticationService
Display form login	Permette la visualizzazione del form di login.	i_authentication_service.IAuthenticationService
Login	Permette il login alla web application SearchQS.	i_authentication_service.IAuthenticationService
Logout	Permette il logout dalla web application SearchQS.	i_authentication_service.IAuthenticationService

4.2. Sottosistema user area

Servizio	Descrizione	Interfaccia
Display user area	Permette la visualizzazione dell'area utente.	i_user_area_service.IUserAreaService
Display form deletion account	Permette la visualizzazione del form per l'eliminazione dell'account.	i_user_area_service.IUserAreaService
Deletion account	Permette l'eliminazione dell'account.	i_user_area_service.IUserAreaService
Display personal data	Permette la visualizzazione dei dati personali.	i_user_area_service.IUserAreaService
Modification personal data	Permette la modifica dei dati personali.	i_user_area_service.IUserAreaService

4.3. Sottosistema analysis

Servizio	Descrizione	Interfaccia
Display analysis area	Permette la visualizzazione dell'area utente.	i_analysis_service.IAnalysisService
Display form loading q. system	Permette la visualizzazione del form per caricare un sistema quantistico.	i_analysis_service.IAnalysisService
Loading q. system	Permette il caricamento di un sistema quantistico.	i_analysis_service.IAnalysisService
Execution analyses	Permette l'esecuzione di uno o più analisi su un sistema quantistico appena caricato.	i_analysis_service.IAnalysisService
Display names transpilation	Permette la visualizzazione dei nomi delle transpilazioni offerte dalla web application SearchQS.	i_analysis_service.IAnalysisService
Display analyses transpilation selected	Permette la visualizzazione di una tabella contenente le analisi eseguite per un tipo di transpilazione o per nessuna transpilazione.	i_analysis_service.IAnalysisService
Display analysis	Permette la visualizzazione completa di una analisi eseguita su un sistema quantistico.	i_analysis_service.IAnalysisService
Display form deletion analysis	Permette la visualizzazione del form per eliminare una analisi selezionata.	i_analysis_service.IAnalysisService
Deletion analysis	Permette l'eliminazione di una analisi selezionata.	i_analysis_service.IAnalysisService

5 Glossario

Nella presente sezione sono raccolti le sigle o i termini del documento che necessitano di una definizione.

Sigla / Termine	Definizione
QSmell	Nome del sistema attuale.
SearchQS	Nome del sistema che si sta realizzando.
COTS	Commercial Off The Shelf, si riferisce a componenti hardware e software disponibili sul mercato per l'acquisto da parte di aziende di sviluppo interessate a utilizzarli nei loro progetti.
Design Goal	Una qualità che il sistema vuole ottenere. Ogni Design Goal si basa su uno o più Requisiti Non Funzionali definiti nel RAD.
Sottosistema	Parte del sistema che raggruppa uno o più Requisiti Funzionali del sistema.
Component Diagram	Diagramma UML che mostra le dipendenze tra i vari sottosistemi.
Diagramma architetturale	Organizzazione di base del sistema espressa dalle sue componenti e dalle relazioni tra di loro e con l'ambiente.
Mapping hardware / software	Studio della connessione tra parti fisiche e logiche di cui si compongono il sistema.