

SYSTEM DESIGN DOCUMENT

Sommario

1. Introduzione	1
1.1 Obiettivi del sistema	
1.2 Design Goals	
1.2.1 Criteri di performance	
1.2.2 Criteri di affidabilità	
1.2.3 Criteri di costo	
1.2.4 Criteri di manutenzione	
1.2.5 Criteri utenti finali	
1.2.6 Trade-off	
1.3 Definizioni, acronimi e abbreviazioni	
1.4 Riferimenti	
1.5 Panoramica	
2. Architettura del Sistema corrente	5
3. Architettura del Sistema proposto	5
3.1 Panoramica	
3.2 Decomposizione in sottosistemi	
3.2.1 Schema Generale	
3.2.2 Gestione Utente	
3.2.3 Gestione Corsi Insegnamento	
3.2.4 Gestione Lezioni	
3.2.5 Gestione Domande	
3.2.6 Diagramma di Deploymen	
3.3 Mapping hardware/software	
3.4 Gestione dei dati persistenti	
3.4.1 Schema ER	
3.4.2 Dizionario dei Dati	
3.4.3 Tavola dei volumi	
3.4.4 Modello logico	
3.5 Controllo degli accessi e sicurezza	
3.6 Controllo flusso globale del sistema	
3.7 Condizioni limite	
4. Servizi dei Sottosistemi.....	22
5. Glossario	24

1. Introduzione

1.1 Obiettivi del sistema

Con l'obiettivo di ottimizzare il servizio di formazione offerto dall'Università degli Studi di Salerno, intendiamo sviluppare un progetto che si collochi nell'ambito dei corsi di studio della Facoltà di Informatica. Lo scopo principale è quello di fornire uno strumento di interazione con gli studenti, utile per lo scambio di domande e valutazioni relative ai vari argomenti spiegati a lezione in modo tale da migliorare sempre di più la didattica all'interno dei vari insegnamenti.

L'obiettivo principale del sistema sarà quello di dare la possibilità allo studente di porre delle domande al docente sui vari argomenti spiegati a lezione, inoltre ci sarà la possibilità di inserire una valutazione personale sulla spiegazione dell'argomento, strumento utile al docente per impegnarsi al meglio nelle lezioni successive. Un ulteriore servizio offerto da Unisask sarà quello di una sezione contenente domande già risposte AQ (Asked Question) dai docenti negli anni precedenti. Quest'ultima funzionalità sarà utile a tutti gli studenti iscritti alla piattaforma per poter approfondire ulteriori argomenti di diversa natura o più semplicemente per poter trovare risposte a dubbi o perplessità già riscontrati negli anni precedenti dagli studenti dello stesso insegnamento. Punto di forza di tale sistema saranno il dialogo tra docente e studente, cosa che al giorno di oggi è molto difficile da instaurare, e la valutazione di ogni singola lezione da parte dell'intera classe, con l'utilizzo di questa piattaforma sarà molto più facile interagire e comunicare tra di loro.

1.2 Design Goals

Il sistema fornirà una struttura chiara, completa e semplice. L'utente non dovrà necessariamente effettuare operazioni che richiedono una conoscenza dell'applicazione, poiché l'utilizzo del sistema da parte dell'utente sarà guidato dall'interfaccia semplice e intuitiva.

L'interfaccia grafica sarà curata nei minimi dettagli, mediante l'utilizzo di bottoni, finestre di dialogo non invasive, label semplici e icone dettagliate e chiare, offrendo così all'utente un'esperienza di utilizzo rapida e semplificata anche per utenti meno esperti. Il sistema proposto rispetterà i criteri di design elencati di seguito.

1.2.1 Criteri di performance

Il sistema garantirà buone performance gestendo adeguatamente tutti gli utenti senza provocare ripetitivi rallentamenti. UniQuestions rispetterà i seguenti requisiti:

- **Tempo di risposta (priorità medio-alta):** Il sistema deve garantire un tempo di risposta inferiore ai 4 secondi soprattutto per le funzionalità stimate tra quelle più frequenti come il login. [Requisito non Funzionale: **RNF_3: Prestazioni**].
- **Throughput (priorità media):** Il sistema dovrà completare il maggior numero possibile di operazioni nel minor tempo possibile. [Requisito non Funzionale: **RNF_3: Prestazioni**].
- **Memoria (priorità medio -bassa):** La dimensione complessiva del sistema dipende dalla memoria utilizzata per il mantenimento del database.

1.2.2 Criteri di affidabilità

UniQuestions garantirà il corretto funzionamento, gestendo vari tipi di errori che potranno verificarsi durante l'utilizzo ed eventuali attacchi alla sicurezza. Quindi, rispetterà i seguenti requisiti di qualità, relativi all'affidabilità:

- **Robustezza (priorità alta):** nel caso in cui l'utente inserisca dati errati nel sistema, quest'ultimo farà visualizzare dei messaggi di errore avvisando l'utente che i dati non sono validi. [Generalizzazione del Requisito non Funzionale: **RNF_2: Affidabilità**].
- **Affidabilità (priorità alta):** Il sistema garantisce il corretto svolgimento delle proprie funzionalità, producendo sempre l'output desiderato. [Requisito non Funzionale: **RNF_2: Affidabilità**].
- **Tolleranza ai guasti (priorità alta):** Il sistema garantisce una tolleranza media agli errori, qualora si verificassero esso dovrà essere in grado di gestirli e risolverli nel minor tempo possibile. [Dedotto dal dominio di applicazione].
- **Security (priorità media):** L'accesso al sistema sarà controllato da un apposito sistema di autenticazione, che permetterà ad ogni utente di accedere alla propria area senza modificare le operazioni altrui, garantendo la privacy. In questo modo nessun utente potrà visualizzare i dati sensibili e accedere alle funzionalità degli altri utenti.

1.2.3 Criteri di costo

- **Costi di sviluppo (priorità media):** È stimato un costo complessivo di 300 ore per la progettazione e lo sviluppo del sistema (50 ore per ogni team member).

1.2.4 Criteri di manutenzione

Il sistema sarà facilmente manutenibile e rispetta i seguenti requisiti:

- **Estendibilità (priorità media):** sistema sarà realizzato in maniera tale da garantire l'aggiunta di nuove funzionalità in maniera semplice. [Generalizzazione del Requisito non Funzionale: **RNF_5: Manutenibilità**].
- **Tracciabilità dei requisiti (priorità alta):** Grazie ad una buona e coerente documentazione, sarà facile risalire ai rispettivi requisiti funzionali, use case e altri artefatti, a cui fanno riferimento le varie classi e metodi del sistema. [Documento RAD].
- **Portabilità (priorità media):** Il sistema sarà portabile in quanto l'interazione avviene mediante un browser senza interazione con il sistema sottostante, c'è quindi indipendenza dal sistema operativo.

1.2.5 Criteri utenti finali

Dal punto di vista dell'utente, il sistema dovrà garantire i seguenti requisiti di qualità:

- **Usabilità (priorità alta):** Il sistema dovrà essere intuitivo e di semplice utilizzo, e sarà progettato tenendo conto di quella che è l'user experience degli studenti. Non sarà necessario l'uso di un manuale utente per compiere le azioni. [Requisito non Funzionale: **RNF_1: Usabilità**].
- **Utilità (priorità alta):** attraverso l'attività di raccolta dei requisiti, il sistema sarà in grado di soddisfare le esigenze degli utenti. [dedotto dal dominio di applicazione].

1.2.6 Trade-off

- **Spazio vs Velocità:** il sistema lavorerà salvaguardando la memoria minimizzando l'utilizzo di essa senza rallentare eccessivamente il sistema.
- **Tempo di rilascio vs Funzionalità:** Verrà rilasciata una prima versione del sistema che comprenderà non tutte le funzionalità ma solo quelle con alta priorità. Le altre funzionalità saranno rilasciate nelle successive release.

1.3 Definizioni, acronimi e abbreviazioni

Acronimi	Descrizione
RAD	Requirement Analysis Document
SDD	System Design Document
Greenfield Engineering	Tipologia di sviluppo che comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall'utente finale e dal cliente.

SQL	Structured Query Language
DBMS	Databse Management System
GUI	Graphical User Interface

1.4 Riferimenti

- RAD_UniQuestions
- Bruegge, Dutoit, Object-Oriented Software Engineering.
- Sommerville, Software Engineering – Addison Wesley.
- Standard StdIEEE 830-1998.

1.5 Panoramica

Il seguente documento è strutturato in quattro parti:

- **Introduzione:** riporta una descrizione del sistema specificando le ragioni del suo sviluppo, le caratteristiche del sistema e un accenno sull'utilizzo delle funzioni.
- **Architettura del sistema corrente:** riporta una breve descrizione dell'architettura al momento esistente.
- **Architettura del sistema proposto:** fornisce una panoramica sull'architettura usata per il sistema. I punti riportati per l'architettura del sistema proposto sono: la suddivisione in sottosistemi, il mapping software-hardware, la gestione dei dati persistenti, il controllo degli accessi di sicurezza, flusso di controllo globale e le condizioni limite.
- **Servizi dei sottosistemi:** espone una descrizione dei sottosistemi identificati e per ognuno i servizi offerti.
- **Glossario:** è una raccolta di termini di quest'ambito specifico.

2. Architettura del Sistema corrente

Attualmente non esiste uno strumento di interazione con gli studenti, utile per lo scambio di domande e valutazioni relative ai vari argomenti spiegati a lezione.

Quindi si tratta di un sistema che rientra nel campo della Greenfield Engineering. Infatti, in un Greenfield Project lo sviluppo comincia da zero, non esiste nessun sistema a priori e i requisiti sono ottenuti dall'utente finale e dal cliente. Nasce, perciò, a partire dai bisogni dell'utente.

3. Architettura del Sistema proposto

3.1 Panoramica

L'architettura scelta per il sistema sarà quella Three-layer. L'utente potrà interagire con l'application layer mediante l'interface layer che offrirà diverse interfacce in base alle necessità dell'utente. L'application layer dovrà poi comunicare con il database per la memorizzazione dei dati persistenti. Sul server, risiede un DBMS che si occupa di recuperare, memorizzare ed interrogare i dati presenti nel database, elaborando, quindi, la richiesta dell'utente. L'aspetto della concorrenza di accessi multipli al database sarà pertanto gestito dal DBMS stesso che dovrà evitare eventuali colli di bottiglia. Tale architettura conferisce all'intero sistema una maggiore manutenibilità, scalabilità e riutilizzo e permette di gestire il problema della concorrenza degli accessi ai dati in maniera semplice ed efficace.

3.2 Decomposizione in sottosistemi

I tre livelli relativi all'architettura adottata sono:

- Interface Layer
- Application Logic Layer
- Data Storage Layer

Interface Layer:

- Si occupa di gestire l'interfaccia utente; include gli oggetti boundary con cui l'utente interagisce (finestre, form, pagine web, ecc..).

Application Logic Layer:

- Comprende le componenti logiche (oggetti control e entity), responsabili del corretto funzionamento del sistema, e utilizza il database per effettuare operazioni sui dati.

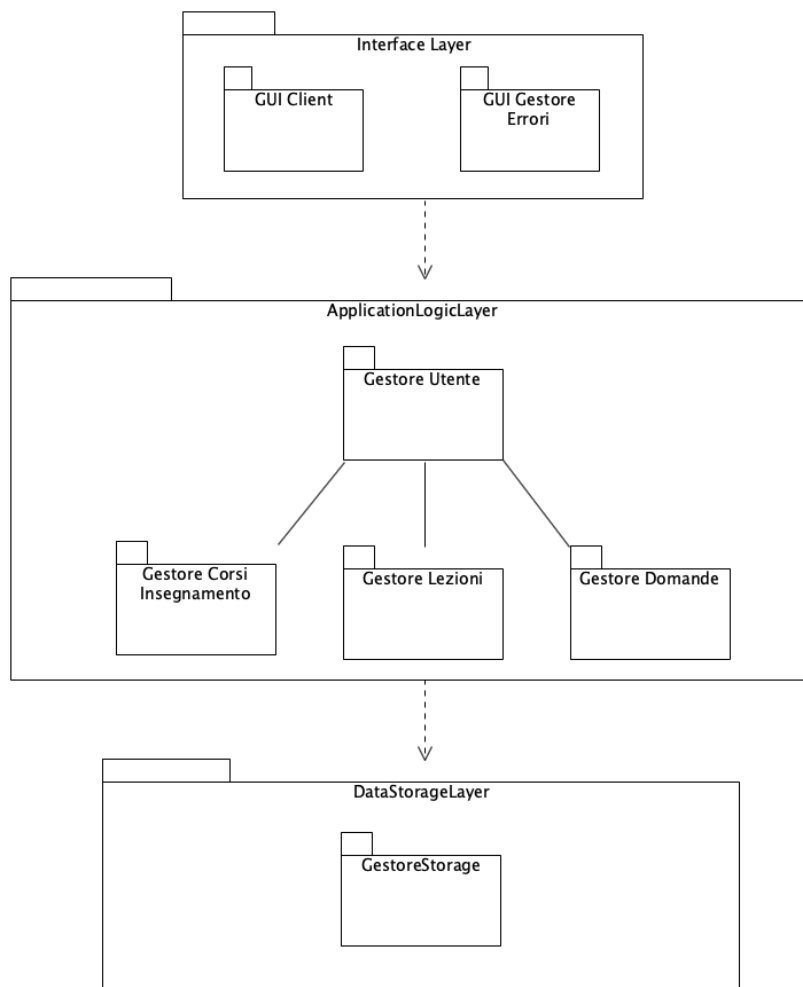
Data Storage Layer:

- Comprende un gestore che si occupa di effettuare operazioni di inserimento, cancellazione e aggiornamento e si occupa di rendere disponibili i dati presenti all'interno del database.

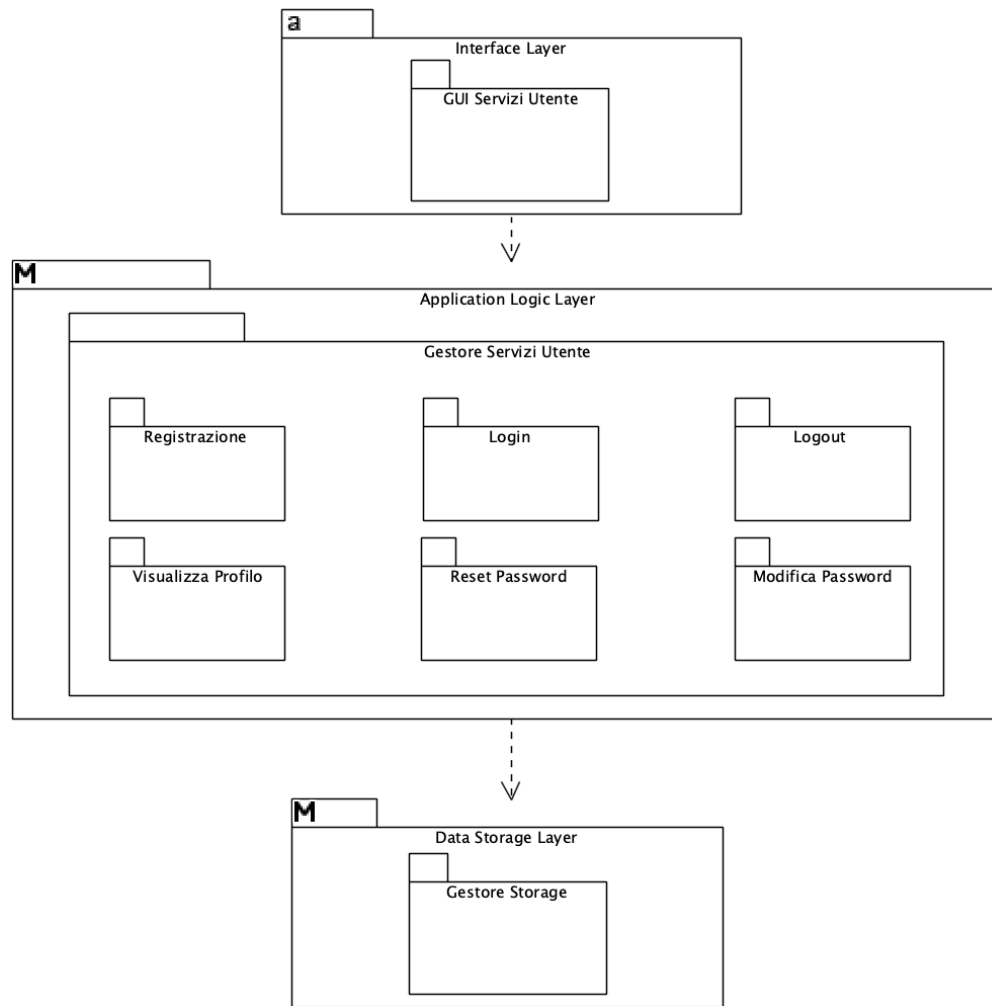
3.2.1 Schema Generale

Lo schema generale è composto dai seguenti sottosistemi mentre di seguito è riportato lo schema generale della suddivisione.

- **Interface Layer:** GUIClient e GUIGestoreErrori, indicano i sottosistemi che contengono tutti gli oggetti boundary.
- **Application Logic Layer:** contiene i quattro sottosistemi individuati (Gestore Corso, Gestore Domande, Gestore Utente e Gestore Lezioni).
- **Gestore Storage:** sottosistema che ha il compito di effettuare operazioni verso il database.



3.2.2 Gestione Utente



Interface Layer:

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per l'accesso ad Unisask:

- **GUI Servizi Utente:** comprende tutte le interfacce dei servizi offerti all'utente.

Application Logic Layer:

Il sottosistema è decomposto in un unico sottosistema:

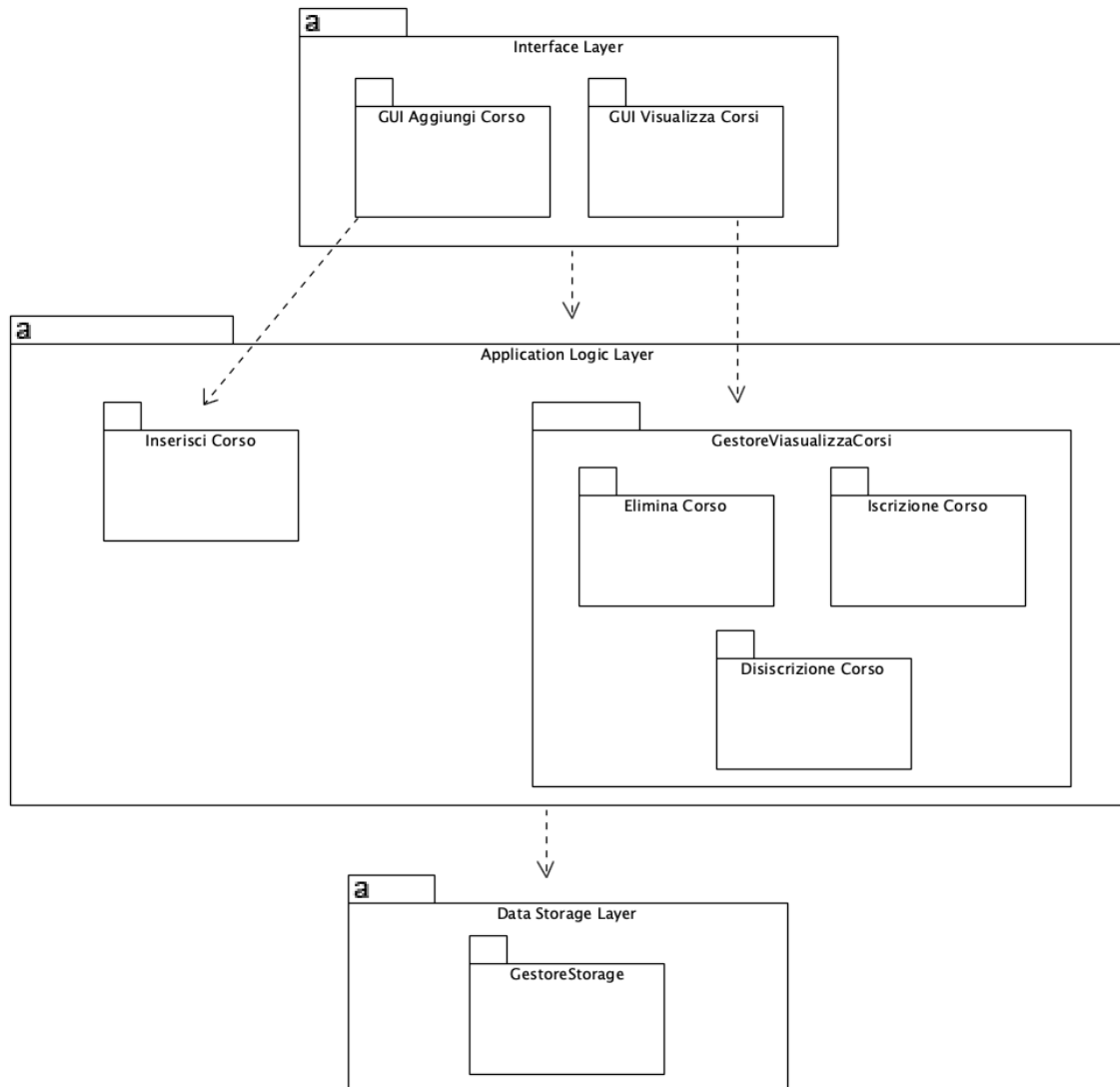
- **Gestore Servizi Utente:** sottosistema che contiene le funzionalità di accesso al sistema quali:
 - **Registrazione**
 - **Login**
 - **Logout**
 - **Visualizza Profilo**
 - **Modifica Password**

- **Reset Password**

Data Storage Layer

Comprende il **Gestore Storage** che permette di effettuare operazioni sul database.

3.2.3 Gestione Corsi Insegnamento



Interface Layer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione dei corsi d'insegnamento:

- **GUI Aggiungi Corso:** comprende l'interfaccia per l'aggiunta di un corso d'insegnamento
- **GUI Visualizza Corsi:** comprende l'interfaccia per la visualizzazione dei corsi d'insegnamento.

Application Logic Layer

Il sottosistema è decomposto in un sottosistema:

- **Inserisci Corso:** permette di aggiungere un nuovo corso d'insegnamento.

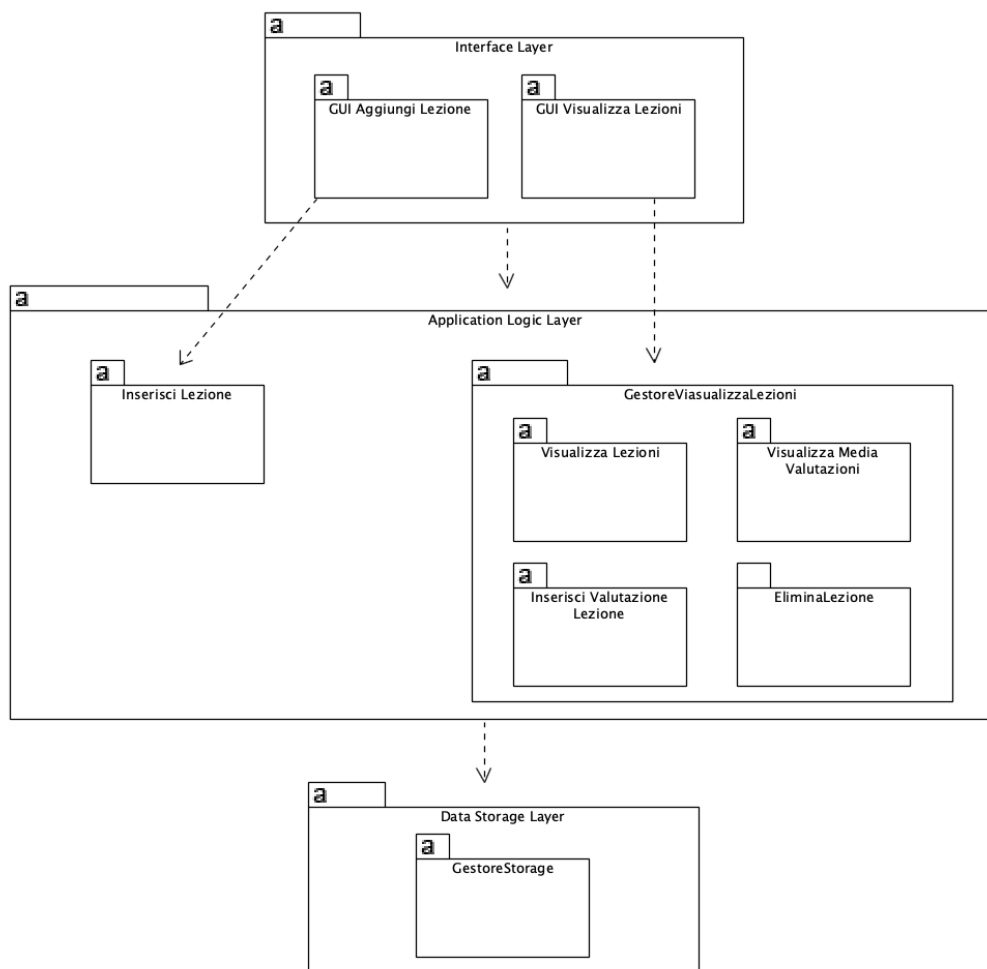
E in un ulteriore sottosistema

- **Gestore Visualizza Corsi:** contiene le funzionalità che offre la visualizzazione dei corsi quali:
 - **Iscrizione Corso**
 - **Disiscrizione Corso**
 - **Elimina Corso**

Data Storage Layer

Comprende il **Gestore Storage** che permette di effettuare operazioni sul database.

3.2.4 Gestione Lezioni



Interface Layer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione delle lezioni:

- **GUI Aggiungi Lezione:** comprende l'interfaccia per l'aggiunta di una lezione.
- **GUI Visualizza lezioni:** comprende l'interfaccia per la visualizzazione delle lezioni.

Application Logic Layer

Il sottosistema è decomposto in un sottosistema:

- **Inserisci Lezione:** permette di aggiungere una nuova lezione.

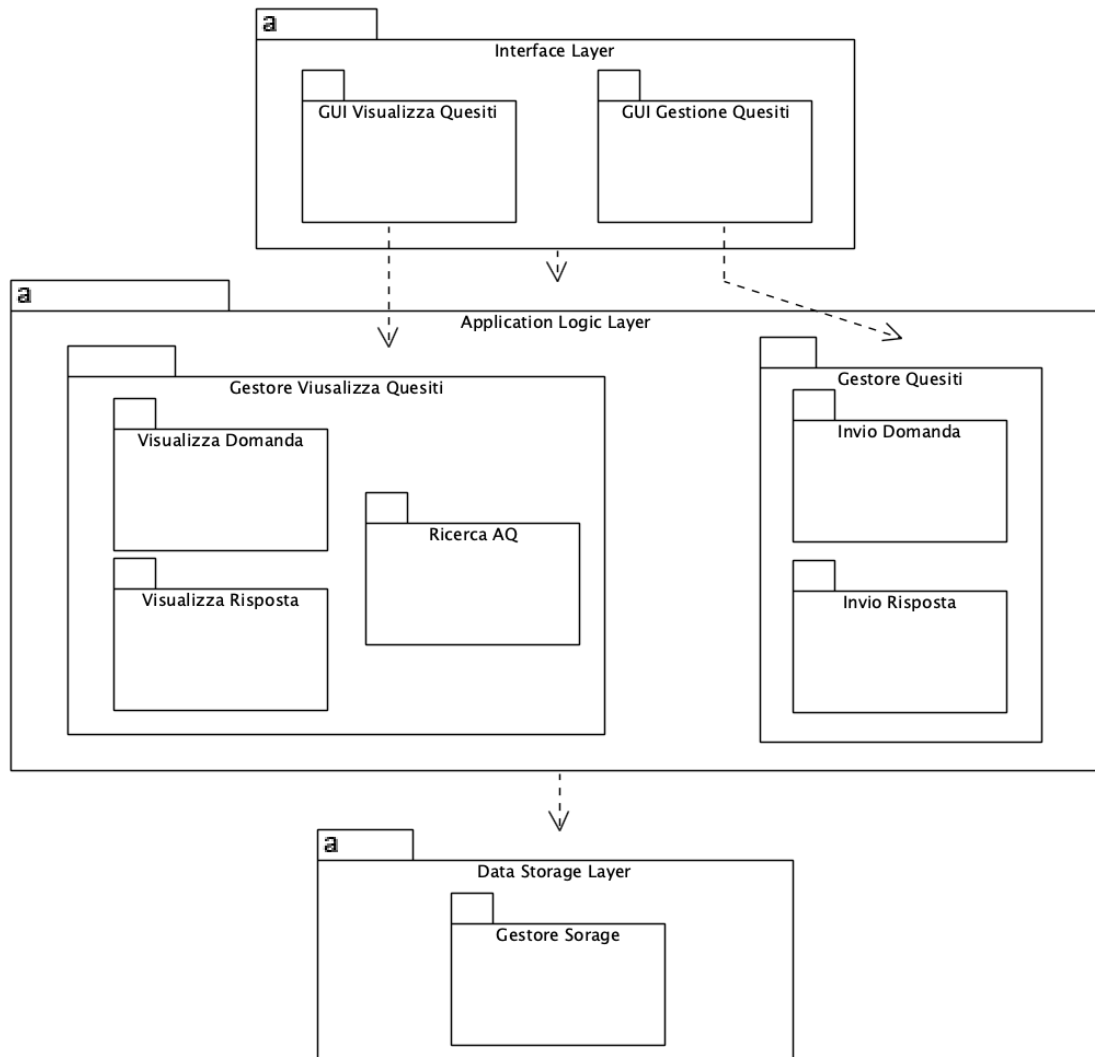
E in un ulteriore sottosistema

- **Gestore Visualizza Lezioni:** contiene le funzionalità che offre la visualizzazione delle lezioni quali:
 - **Elimina Lezione**
 - **Inserisci Valutazione Lezione**
 - **Visualizza Media Valutazioni**
 - **Visualizza Lezioni**

Data Storage Layer

Comprende il **Gestore Storage** che permette di effettuare operazioni sul database.

3.2.5 Gestione Domande



Interface Layer

Include tutte le componenti dell'interfaccia grafica del sistema che offrono le funzionalità per la gestione delle domande:

- **GUI Visualizza Quesiti:** comprende l'interfaccia per la visualizzazione dei quesiti.
- **GUI Gestione Quesiti:** comprende l'interfaccia per la gestione dei quesiti.

Il sottosistema è decomposto in ulteriori due sottosistemi:

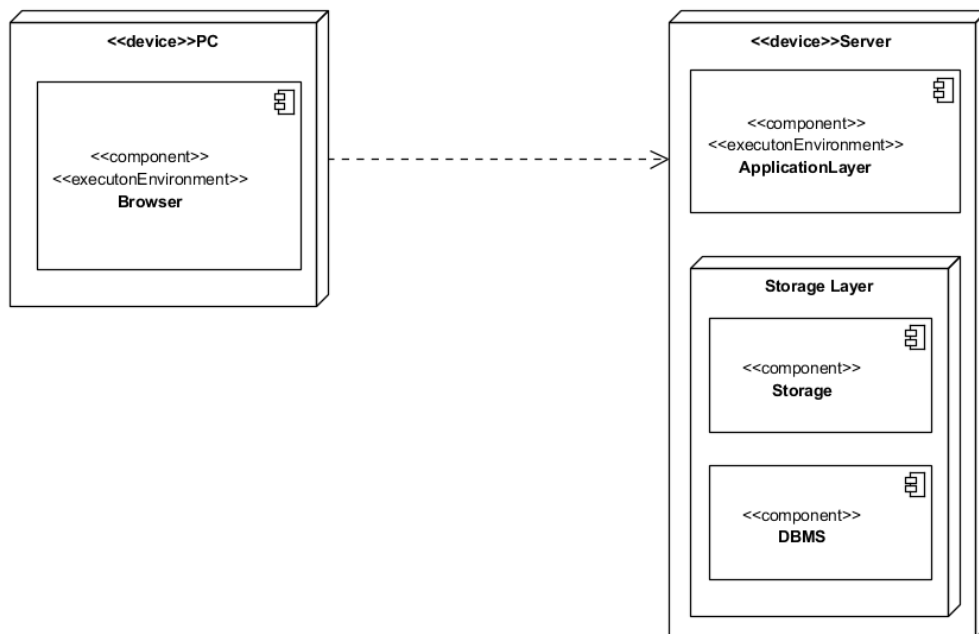
- **Gestore Visualizza Quesiti:** che permette le seguenti funzionalità:
 - **Visualizza Domanda**
 - **Visualizza Risposta**
 - **Ricerca AQ**
- **Gestore Quesiti:** che permette le seguenti funzionalità:
 - **Invio Domanda**
 - **Invio Risposta**

Data Storage Layer

Comprende il Gestore Storage che permette di effettuare operazioni sul database.

3.2.6 Diagramma di Deployment

Il diagramma di deployment mostra le componenti che utilizzerà il nostro sistema. Questo diagramma aiuterà gli sviluppatori a comprendere le relazioni tra le componenti software e i nodi hardware. Al lato client, l'interfaccia utente verrà mostrata sul browser web ed interagirà con l'application logic layer, che a sua volta memorizza e interroga i dati persistenti sul database.



3.3 Mapping hardware/software

Il sistema sviluppato sarà sviluppato su un solo server e utilizzerà un DBMS MySql stanziato sullo stesso per la gestione dei dati persistenti. Il sistema sarà diviso in client e server che inizialmente saranno lo stesso elaboratore, ma in caso di futura necessità potranno essere facilmente divisi in quanto i servizi saranno progettati separatamente. Sul server ci sarà un DBMS per la gestione dei dati persistenti di nostro interesse e il client a cui offre servizi potrà essere un qualunque pc che abbia un browser installato.

3.4 Gestione dei dati persistenti

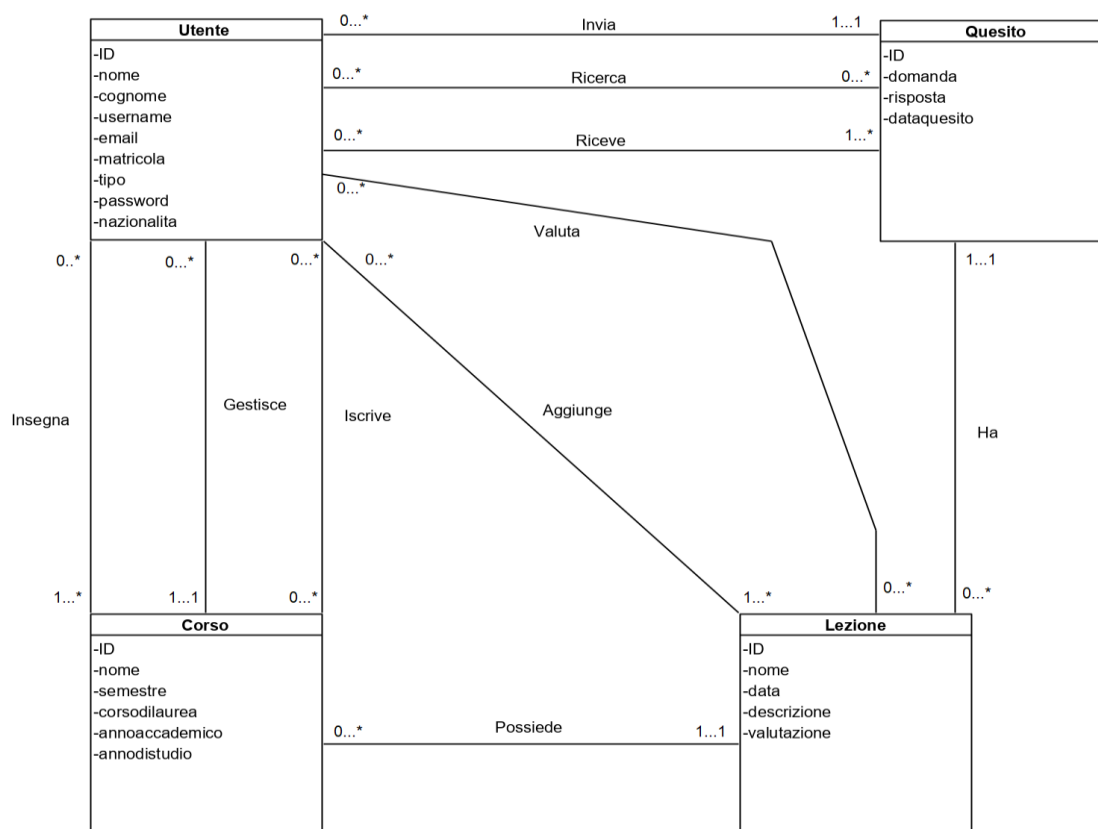
Il sistema userà un servizio di storage su Database. Le risorse saranno opportunamente salvate in cartelle sul server. La scelta del DBMS è ricaduta su MySQL viste le conoscenze dei team member. La scelta di un database relazionale, rispetto a un database object-oriented, è motivata dalle alte prestazioni offerte dai primi.

Legenda priorità:

Urgentissimi	
Consigliato inserire	
Trascurabili	

NOME INDICATORE	DATI	PRIORITÀ
Utente	<ul style="list-style-type: none">- ID- Nome- Cognome- Username- E-mail- Password- Tipo	
Utente	<ul style="list-style-type: none">- Nazionalità- Matricola	
Corso	<ul style="list-style-type: none">- ID- Nome- Anno accademico- Corso di laurea- Semestre- Anno di studio	
Lezione	<ul style="list-style-type: none">- ID- Nome- Valutazione	
Lezione	<ul style="list-style-type: none">- Data- Descrizione	
Quesito	<ul style="list-style-type: none">- ID- Risposta- Domanda	
Quesito	<ul style="list-style-type: none">- Data quesito	

3.4.1 Schema ER



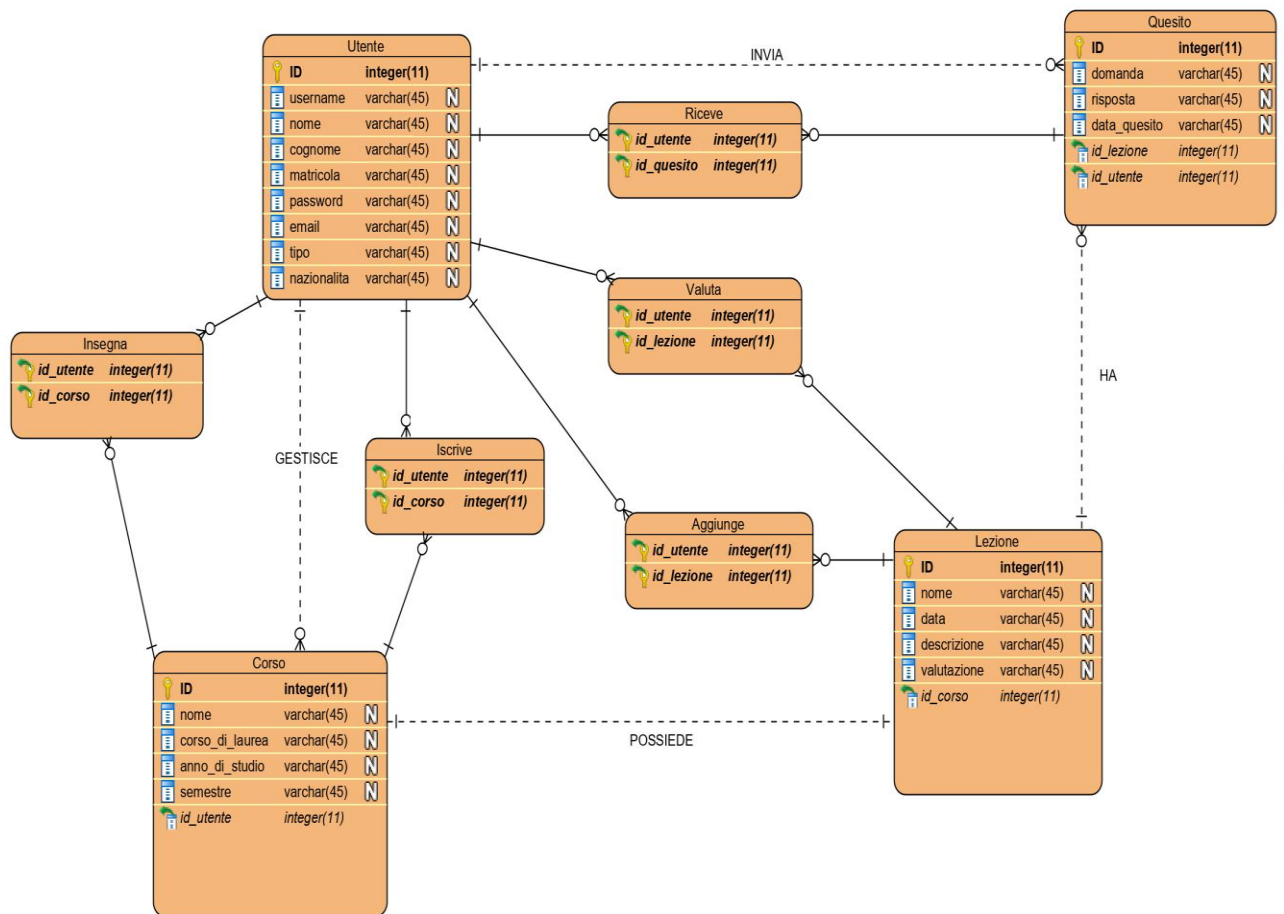
3.4.2 Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Utente	Elenco degli utenti registrati al sistema	username, password, nome, cognome, e-mail, matricola, nazionalita	ID
Corso	Elenco dei corsi creati dall'amministratore.	corso_dilaurea, annoaccademico, annodistudio, semestre, nome	ID
Lezione	Elenco delle lezioni inserite nei corsi	nome, data, descrizione, valutazione	ID
Quesito	Elenco dei quesiti posti dagli studenti	domanda, risposta, dataquesito	ID

3.4.3 Tavola dei volumi

Concetto	Costrutto	Volume
Utente	Entità	4000
Corso	Entità	200
Lezione	Entità	350
Quesito	Entità	400
Gestisce	Relazione	200
Insegna	Relazione	50
Riceve	Relazione	400
Aggiunge	Relazione	200
Iscrive	Relazione	900
Ricerca	Relazione	400
Invia	Relazione	400
Valuta	Relazione	350
Possiede	Relazione	350
Ha	Relazione	400

3.4.4 Modello logico



3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi sarà alla base della sicurezza del nostro sistema, permettendo ad ogni utente o amministratore di collegarsi al sistema tramite l'utilizzo di username e password, che verranno richieste ad ogni singolo accesso. La sessione termina quando l'utente effettua il logout.

Nel caso l'accesso al sistema non abbia successo, verrà visualizzata una notifica di fallimento indicando che è avvenuto un errore nell'inserimento di username o password e consentendo all'utente di effettuare un nuovo tentativo.

Caratteristica di Unisask è quella di non prevedere nessuna funzionalità ad utenti che non sono registrati. I visitatori collegandosi alla piattaforma potranno soltanto scegliere di registrarsi oppure essendo già registrati di effettuare l'accesso.

Il sistema fornirà più viste (interfacce grafiche) a seconda dell'attore che ci interagirà, in modo che ognuno possa accedere solo alle rispettive funzionalità previste in modo tale da rafforzare la sicurezza dei dati sensibili di ogni utente.

Sottosistema Attore	Gestione			
	Gestione Utente	Gestione Corsi Insegnamento	Gestione Lezioni	Gestione Domande
Amministratore	<ul style="list-style-type: none"> Login Logout 	<ul style="list-style-type: none"> Inserisci Corso Insegnamento Elimina Corso Insegnamento Visualizza Corsi Insegnamento 	NA	NA
Studente	<ul style="list-style-type: none"> Registrazione Login Logout Reset password Modifica Password Visualizza Profilo 	<ul style="list-style-type: none"> Iscrizione Corso Disiscrizione Corso 	<ul style="list-style-type: none"> Visualizza Lezioni Inserisci Valutazione Lezione 	<ul style="list-style-type: none"> Invio Domanda Visualizza Risposta Ricerca AQ
Docente	<ul style="list-style-type: none"> Registrazione Login Logout Reset Password Modifica Password Visualizza Profilo 	<ul style="list-style-type: none"> Visualizza Corsi 	<ul style="list-style-type: none"> Aggiungi Lezione Elimina Lezione Visualizza Media Valutazioni 	<ul style="list-style-type: none"> Invia Risposta Visualizza Domande

3.6 Controllo flusso globale del sistema

Per quanto riguarda il flusso di controllo esterno fra sottosistemi, il server sarà sempre in funzione (a meno di manutenzioni) in attesa di eventuali richieste di servizi da parte dell'utenza; il sistema non avrà problemi a gestire più utenti contemporaneamente, ma nel caso di molte richieste contemporanee, il server potrebbe ovviare al sovraccarico utilizzando la tecnica FIFO.

Quindi il controllo del flusso globale del sistema sarà di tipo event-driven.

3.7 Condizioni limite

3.7.1 Fallimento

Possono verificarsi diversi casi del fallimento del sistema:

1. Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che ripristinino lo stato del sistema a prima dello spegnimento inaspettato.
2. Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione, non sono previste politiche correttive, l'unico processo che potrà essere eseguito è la chiusura del sistema e il suo successivo riavvio.
3. Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, non è prevista alcuna misura correttiva.
4. Link ad una pagina inesistente: il sistema mostrerà all'utente una pagina che lo avvisa dell'errore e gli permetterà di tornare alla homepage.
5. Durante la registrazione di un nuovo utente se dei campi non verranno completati correttamente la registrazione fallirà, mostrando un messaggio d'errore.
6. Nell'inserimento di una nuova lezione o corso nel sistema, se i dati non verranno inseriti correttamente, l'inserimento fallirà, non inserendo i dati nel database del sistema, e mostrando un messaggio di errore del mancato inserimento/caricamento.

3.7.2 Use Case Condizioni Limite

Di seguito sono riportate alcune condizioni limite a cui il sistema può andare incontro.

3.7.2.1 UC_CL_1 Start Server

Identificativo UC_CL_1	StartServer	Data	02/12/2018
		Vers.	0.1.0
		Autore	Antonio Vitiello
Descrizione	Lo UC fornisce la funzionalità di avviare il server		
Attore Principale	Amministratore: Vuole avviare il server.		
Attori secondari	NA		
Entry Condition	Il server è spento e funzionante		

Exit condition On success		Il server viene avviato con successo
Exit condition On failure		Il server non viene avviato con successo
Rilevanza/User Priority		ALTA
Frequenza stimata		5/anno
Extension point		NA
Generalization of		NA
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Amministratore:	Avvia il server
2	Server:	Segnala che l'avvio è avvenuto con successo e i suoi servizi vengono messi a disposizione degli utenti
I Scenario/Flusso di eventi di ERRORE: Il server non viene avviato con successo		
2.1	Server:	Segnala che non è stato possibile avviare il server

3.7.2.2 UC_CL_2 Shutdown Server

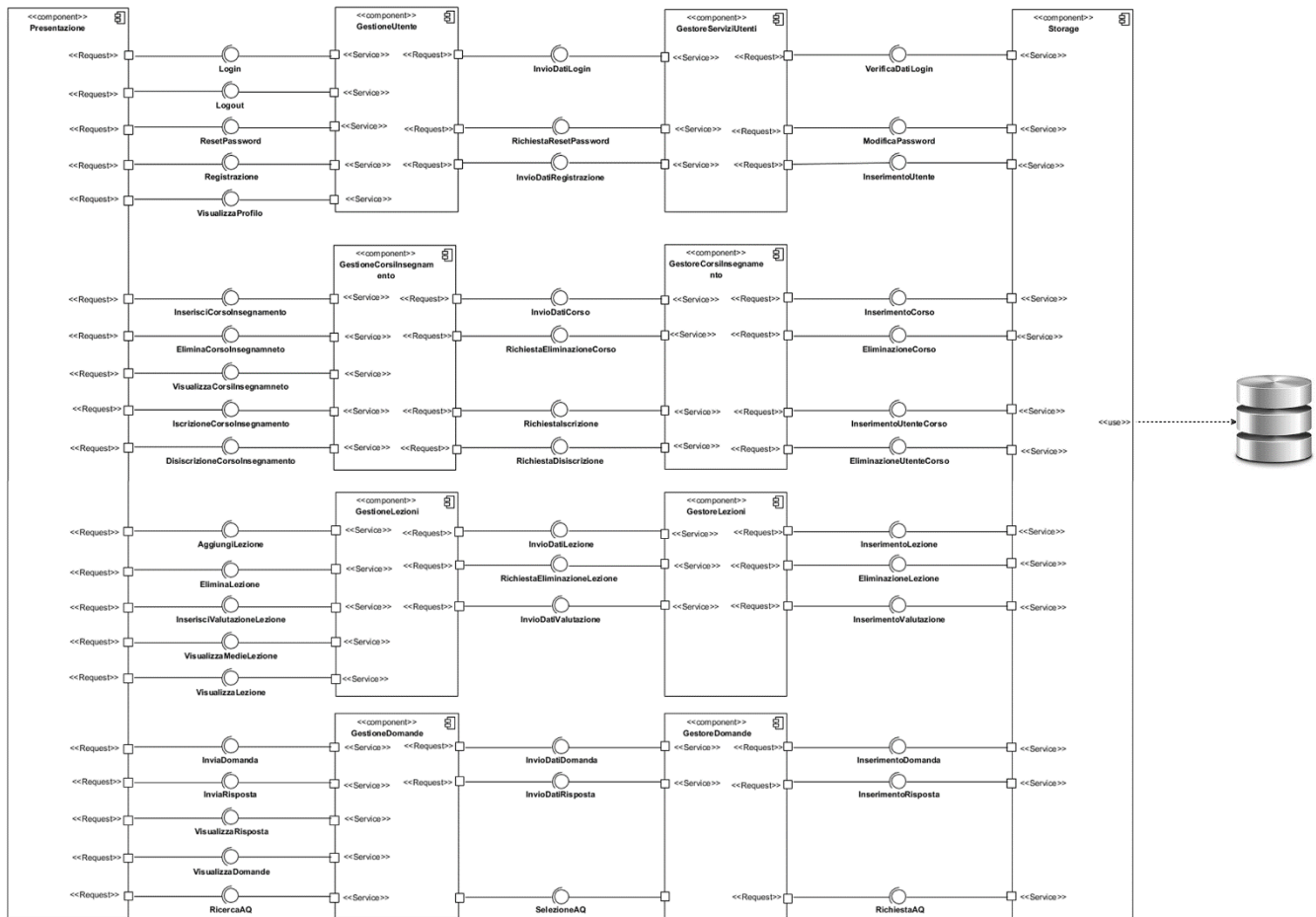
Identificativo UC_CL_2	ShutdownServer	Data	02/12/2018
		Vers.	0.1.0
		Autore	Antonio Vitiello
Descrizione	Lo UC fornisce la funzionalità di arrestare il server		
Attore Principale	Amministratore: vuole arrestare il server		
Attori secondari	NA		
Entry Condition	Il server è acceso		
Exit condition On success	Il server viene arrestato		
Exit condition On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	5/anno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Amministratore:	Spegne il server	

2	Server:	Si spegne con successo
---	---------	------------------------

3.7.2.3 UC_CL_3 Link ad una pagina inesistente

Identificativo UC_CL_3	Link ad una pagina inesistente	Data	02/12/2018
		Vers.	0.1.0
		Autore	Antonio Vitiello
Descrizione	L'utente tenta di accedere ad un link di una pagina inesistente		
Attore Principale	Utente		
Attori secondari	NA		
Entry Condition	L'utente tenta di accedere ad un link di una pagina inesistente		
Exit condition On success	Viene visualizzata una pagina di errore		
Exit condition On failure	NA		
Rilevanza/User Priority	BASSA		
Frequenza stimata	5/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	Cerca di accedere ad una pagina inesistente	
2	Sistema:	Mostra una pagina di errore, indicando che la pagina non esiste.	

4. Servizi dei Sottosistemi



Presentazione: Interfaccia che gestisce l'interazione tra Utente e sistema.

GestioneUtente offre 5 servizi all'interfaccia Presentazione:

- Login
- Logout
- ResetPassword
- Modifica Password
- Registrazione
- VisualizzaProfilo

GestioneCorsoInsegnamento offre 5 servizi all'interfaccia Presentazione:

- InserisciCorsoInsegnamento
- EliminaCorsoInsegnamento

- VisualizzaCorsiInsegnamento
- IscrizioneCorsoInsegnamento
- DisiscrizioneCorsoInsegnamento

GestioneLezioni offre 5 servizi all'interfaccia Presentazione:

- AggiungiLezione
- EliminaLezione
- InserisciValutazioneLezione
- VisualizzaMedieLezione
- VisualizzaLezione

GestioneDomande offre 5 servizi all'interfaccia Presentazione:

- InviaDomanda
- InviaRisposta
- VisualizzaRisposta
- VisualizzaDomanda
- RicercaAQ

GestoreServiziUtenti offre 3 servizi all'interfaccia GestioneUtente:

- InviaDatiLogin
- RichiestaResetPassword
- InvioDatiRegistrazione

GestoreCorsoInsegnamento offre 4 servizi all'interfaccia GestioneCorsoInsegnamento:

- InviaDatiCorso
- RichiestaEliminazioniCorso
- RichiestaIscrizione
- RichiestaDisiscrizione

GestoreLezione offre 3 servizi all'interfaccia GestioneLezione:

- InvioDatiLezione
- RichiestaEliminazioneLezione
- InvioDatiValutazione

GestoreDomande offre 5 servizi all'interfaccia GestioneDomande:

- InvioDatiDomanda
- InvioDatiRisposta
- SelezioneAQ

Storage: offre i servizi a (Gestore Servizi Utente, Gestore Corso Insegnamento, Gestore Lezione, Gestore Domande):

- VerificaDatiLogin
- ModificaPassword
- InserimentoUtente
- InserimentoCorso
- EliminazioneCorso
- InserimentoUtenteCorso
- EliminazioneUtenteCorso
- InserimentoLezione
- EliminazioneLezione
- InserimentoValutazione
- InserimentoDomanda
- InserimentoRisposta
- RichiestaAQ

5. Glossario

- **Design goals:** Obiettivi qualitativi del sistema, identificano le qualità su cui deve essere focalizzato il sistema.

- **Trade-offs:** Compromessi.
- **Persistent data management:** Strategie relative alla gestione dei dati persistenti.
- **Schema ER:** un modello per la rappresentazione concettuale dei dati ad un alto livello di astrazione.
- **Dizionario dei dati:** Permette di arricchire lo schema ER con descrizioni in linguaggio naturale di entità, attributi e relazioni.
- **Tavola dei volumi:** Specifica il numero stimato di istanza per ogni entità e associazione dello schema.
- **Modello logico:** Il modello logico discende dal modello concettuale e disegna un'architettura che tiene conto delle strutture proprie di quel particolare database.
- **Condizioni limite:** Soluzioni a problemi fisici.