



System Design Document Sorting Hat

Riferimento	
Versione	1.4
Data	14/02/2022
Destinatario	Docente Ingegneria del Software 2021/22
Presentato da	Ascione Josef, Di Gregorio Emanuele, Di Sarno Davide, Sacco Matteo
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
17/11/2021	0.1	Prima stesura	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.



Corso Triennale di Informatica – Università di Salerno
 Corso di studi di Ingegneria del Software
 Sorting Hat

19/11/2021	0.2	Modifica design goals	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
26/11/2021	0.3	Revisione	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
27/11/2021	0.4	Modifica	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
01/12/2021	0.5	Modifica Current software	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
3/12/2021	0.6	Raffinamento punti architettura Proposta	Di Sarno D, Ascione J.
05/12/2021	0.7	Modifica	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
06/12/2021	0.8	Modello ER e decomposizione in sottosistemi	Ascione J., Di Gregorio E., Di Sarno D., Sacco M.
10/12/2021	0.9	Modifica modello ER	Ascione J., Di Gregorio E.
11/12/2021	1.0	Revisione Documento	Ascione J., Di Gregorio E.
08/01/2022	1.1	Revisione condizioni limite e architettura proposta	Di Sarno D.,Di Gregorio E.
10/01/2022	1.2	Revisione architettura proposta	Di Sarno D.,Di Gregorio E.
12/02/22	1.3	Modifica gestione della persistenza e strutturazione del Subsystem services	Sacco M.
14/02/22	1.4	Modifica finale e punto 4	Ascione A.,Di Sarno D.



Sommario

Sommario	3
1. Introduction	4
1.1 Scopo del sistema	4
1.2 Design Goals	4
1.2.1. Design trade-offs	6
1.3 Definizioni, Acronimi ed abbreviazioni	7
1.4 Referenze	7
1.5 Panoramica	7
2. Architettura corrente	8
3. Architettura proposta	9
3.1 Panoramica	9
3.2. Subsystem decomposition	9
3.3. Hardware/software mapping	19
3.4. Gestione della Persistenza	20
3.5. Access control and security	23
3.6. Global software control	24
3.7. Boundary conditions	25
4. Subsystems services	29



1.Introduction

1.1 Scopo del sistema

“Sorting Hat” è una piattaforma web utile a far comunicare tra di loro studenti dell’Università degli studi di Salerno e possibili matricole e/o ospiti del sito.

Le discussioni tra studenti avverranno su un Forum **Thread** memorizzati su un server.

La piattaforma supporta, attraverso un **modulo IA**, un servizio di consulting che aiuterà le possibili matricole a scegliere il percorso di studi più adatto a loro.

1.2 Design Goals

In questa sezione si presenteranno i Design Goal, ovvero le qualità sulle quali si focalizza il sistema.

Seguendo le linee guida del libro Bernd Bruegge -Object Oriented Software Engineering , i design goal sono stati suddivisi nelle seguenti categorie:

- **Performance**:includono i requisiti di spazio e velocità imposti sul sistema
- **Dependability**: determinano quanto sforzo deve essere speso per minimizzare i fallimenti del sistema (crash, falle di sicurezza) e le loro conseguenze.
- **Maintenance**:determina quanto sforzo è necessario per modificare il sistema dopo il suo rilascio.
- **End User**:includono qualità che sono desiderabili dal punto di vista dell’utente, ma che non sono state coperte dai criteri di Performance e Dependability.

Ciascun design goal è descritto da:

- **Rank**, che specifica la priorità da 1 (massimo) a 4(minimo);
- **ID Design goal**, un identificatore univoco per il design goal;
- **Descrizione**, descrizione del design goal;
- **Categoria**, ovvero la categoria di appartenenza del design goal.

RANK	ID DESIGN GOAL	DESCRIZIONE	CATEGORIA	RNF di origine
1	DG_1 Sicurezza accessi	Il sistema deve suddividere le aree accessibili dai vari utenti a secondo delle restrizioni legate	Dependability	RNF_4



		al loro ruolo all'interno della piattaforma		
2	DG_2 Sicurezza form	Il sistema deve proteggere le informazioni degli utenti, ottenute dai form, da eventuali violazioni.	Dependability	RNF_6
4	DG_3 Evoluzioni future	Il sistema deve potersi adattare facilmente all'uso di possibili nuove tecnologie.	Maintenance	RNF_10
1	DG_4 Facilità d'uso	“Sorting Hat” deve essere di semplice utilizzo senza l'ausilio di manuali da parte dell'utente.	End User	RNF_1
1	DG_5 Accesso ospiti	Il sistema deve garantire agli utenti non registrati di visualizzare il form e i forum senza effettuare l'accesso.	End User	RNF_2
2	DG_6 Navigazione concorrente	Il sistema deve supportare l'accesso in simultanea degli utenti registrati senza dover limitare l'accesso alle operazioni messe a disposizione.	Performance	RNF_7
3	DG_7 Forum concorrente	Il sistema deve permettere agli utenti-ordinari e agli utenti-universitari di	Performance	RNF_8



		pubblicare domande e risposte in parallelo senza creare problemi di concorrenza.		
1	DG_8 Affidabilità	Il sistema deve garantire un funzionamento corretto anche in presenza di input non validi da parte dell'utente e condizioni di stress dovuto a eventuale sovraccollamento del sito.	Dependability	RNF_6
3	DG_9 Manutenzione	Il sistema deve poter garantire una buona manutenzione	Maintenance	RNF_11

1.2.1. Design trade-offs

Design Trade-off:

Spazio vs. Tempi di risposta= Fin quando possibile il sistema deve garantire una maggiore velocità di risposta (per l'elaborazione e la visualizzazione all'utente) a discapito della memoria utilizzata.



Tempi di risposta vs. Affidabilità=Essendo il sistema basato sull'interazione con l'utente, deve preferire una maggiore affidabilità nelle sue operazioni piuttosto che ai tempi di risposta per poter garantire dati di input e output validi.

Tempi di risposta vs. Sicurezza=Per garantire la sicurezza del sito ed evitare operazioni e accessi illegali può essere necessario sacrificare la velocità delle elaborazioni.

Usabilità vs Estensibilità: Se per rendere il sistema estensibile dovremo sacrificare del tempo per quanto riguarda l'usabilità allora potrà essere necessario dare la priorità alla facilità di utilizzo

Sicurezza vs Estensibilità: Se per garantire la sicurezza del sistema, in merito ad accessi a pagine web e l'esecuzione di operazioni soggette a limitazioni relative ai ruoli degli utenti, dovremmo ridurre la facilità di aggiunta di nuove funzionalità per impedire operazioni illegali.

1.3 Definizioni, Acronimi ed abbreviazioni

- “**sezione consiglio**”: Per sezione consiglio intendiamo la parte del sito web nella quale l'utente Ordinario (ossia la matricola) può compilare il form per ricevere consiglio dal modulo di IA.
- “**thread**”: Nel gergo dei forum indica una discussione.
- “**modulo IA**”: Indica il modulo di intelligenza artificiale che opera nel nostro sistema.
- “**sezione contribuisce**”: Per sezione contribuisce intendiamo un'opzione del sito web nella quale un utente universitario può compilare un form riguardo la sua esperienza universitaria per il modulo di IA.
- “**utente ospite**”: Indica l'utente non registrato.
- “**JDBC**”: Java Database Connectivity
- “**DBMS**”: Database Management System
- “**HTTP**”: Hypertext Transfer Protocol

1.4 Reference

- RAD
 - Versione 1.1
- Libro: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition
Autori: Bernd Bruegge & Allen H. Dutoit

1.5 Panoramica

Il documento redatto è suddiviso in sottosezioni ordinate:

- Sezione 2: Current software architecture, descrive l'architettura del sistema tenendo conto di eventuali sistemi simili preesistenti.
- Sezione 3: Proposed software architecture, suddiviso in sottosezioni:
 - 3.1. Overview, breve descrizione delle funzionalità di ogni sottosistema



- 3.2. Subsystem decomposition ,scomposizione in sottosistemi con rispettive responsabilità.
 - 3.3. Hardware/software mapping, associazione tra componenti hardware e off-the-shelf con i vari sottosistemi.
 - 3.4. Persistent data management, analisi dei dati persistenti e utiizzo del database. *
 - 3.5. Access control and security, describe la matrice di accesso utilizzata.
 - 3.6. Global software control, describe l'implementazione del flusso di controllo globale.
 - 3.7. Boundary conditions, describe le condizioni di start-up, shutdown e fallimento del sistema.
- Sezione 4: Subsystem services, describe i servizi offerti da ogni sottosistema.
 - Glossario dei termini

2. Architettura corrente

Attualmente, non essendoci un software delegato a tale ruolo, la conoscenza dell'ambito universitario è relegato ad un continuo passaparola tra conoscenti ed amici di amici.

Possiamo quindi affermare che “Sorting Hat” tenterà di digitalizzare tale pratica affidando il compito di descrivere l'ambiente universitario agli stessi studenti tramite l'ausilio del forum e il compito di consigliare un plausibile corso di studi tramite l'aiuto di un'IA.



3. Architettura proposta

3.1 Panoramica

Il sistema proposto è un'applicazione web basata sull'interazione con gli utenti e con un modulo IA esterno.

Gli utenti sono divisi in varie categorie:

- Universitari, se sono studenti di un percorso universitario
- Ordinari, se sono neodiplomati che cercano informazioni sull'Università
- Admin, se necessita di privilegi specifici per la gestione del sistema

Tutti gli utenti possono effettuare login e logout e svolgere specifiche operazioni.

In particolare, gli studenti universitari e ordinari saranno partecipanti a discussioni in merito agli specifici dipartimenti universitari.

Per la realizzazione del sistema sarà utilizzata un'architettura Three Tier, strutturata in questi 3 elementi:

- **Interface:** si occupa di far visualizzare i dati dall'utente e dalla creazione di pagine dinamiche (JSP)
- **Application:** riceve i comandi dall'utente attraverso le view e si occupa dell'interazione con le altre componenti, per la creazione e l'inoltro della risposta alla View(Servlet)
- **Storage:** Si occupa di rendere i dati persistenti e di effettuare metodi di lettura e scrittura su database (DAO) e metodi riguardanti classi java(Beans).

Essendo un'applicazione web, lo sviluppo utilizzerà HTML5, CSS3 e Figma come tool relativi al front-end e la generazione view.

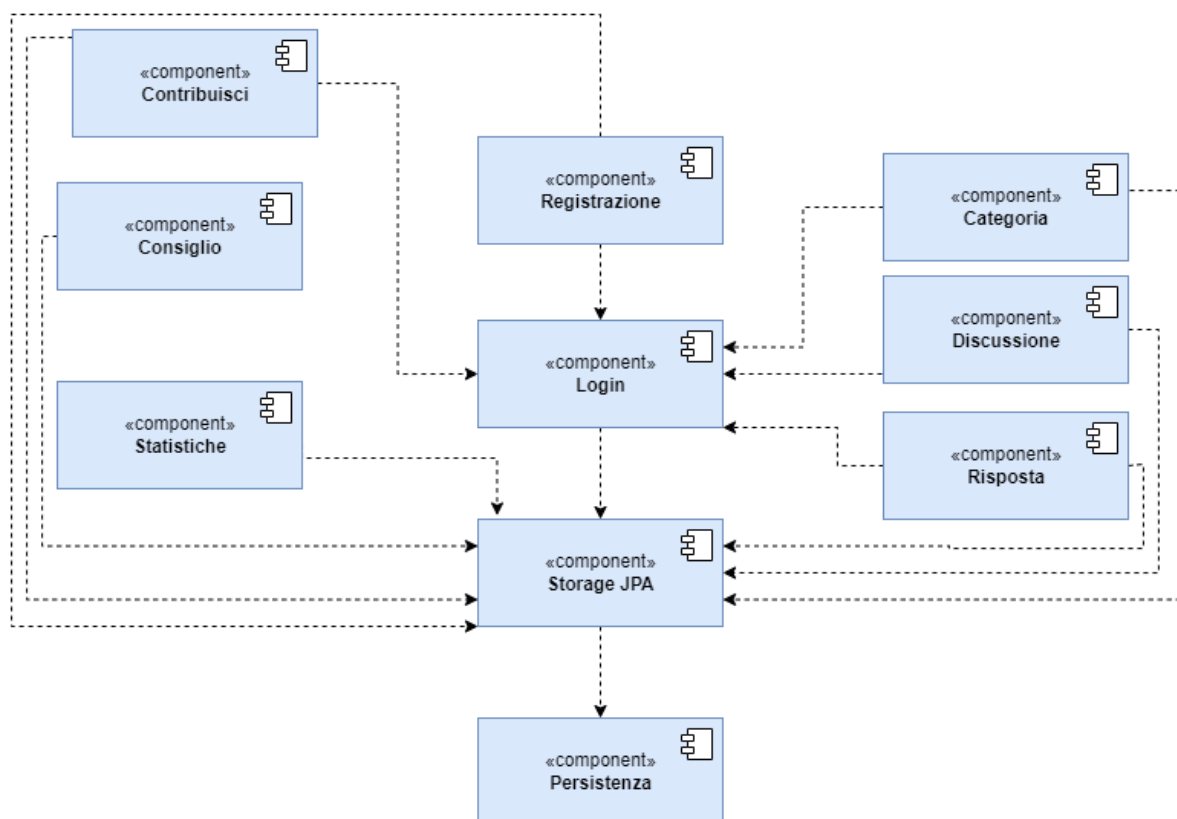
Per la logica di back-end verrà usato Java mentre il database sarà gestito usando SQL per il database su cloud e JDBC per il collegamento.

3.2. Subsystem decomposition

I sottosistemi da noi individuati sono:

- **Registrazione:** gestisce la registrazione dei due diversi tipi di utenza(utenti ordinari ed universitari).
- **Login:** si occupa delle funzioni di: Login, Logout, visualizzazione e modifica del profilo utente.
- **Contribuisci:** gestisce l'invio di form per contribuire all'allenamento del modulo IA.
- **Consiglio:** si occupa sia dell'invio del form al modulo IA e sia della ricezione del messaggio di risposta.
- **Statistiche:** gestisce l'accesso alla pagina per la visualizzazione delle statistiche del sito.
- **Categoria:** si occupa delle funzioni riguardanti le categorie.
- **Discussione:** si occupa delle funzioni riguardanti le discussioni.
- **Risposta:** si occupa delle funzioni riguardanti le risposte.
- **Persistenza:** gestisce la persistenza dei dati tramite un DB.
- **Storage JPA:** fa da tramite tra i vari sottosistemi e la Persistenza.

Di seguito il component diagram UML:



Alcuni sottosistemi saranno gestiti da componenti COTS (Commercial off the shelf), di seguito un elenco:

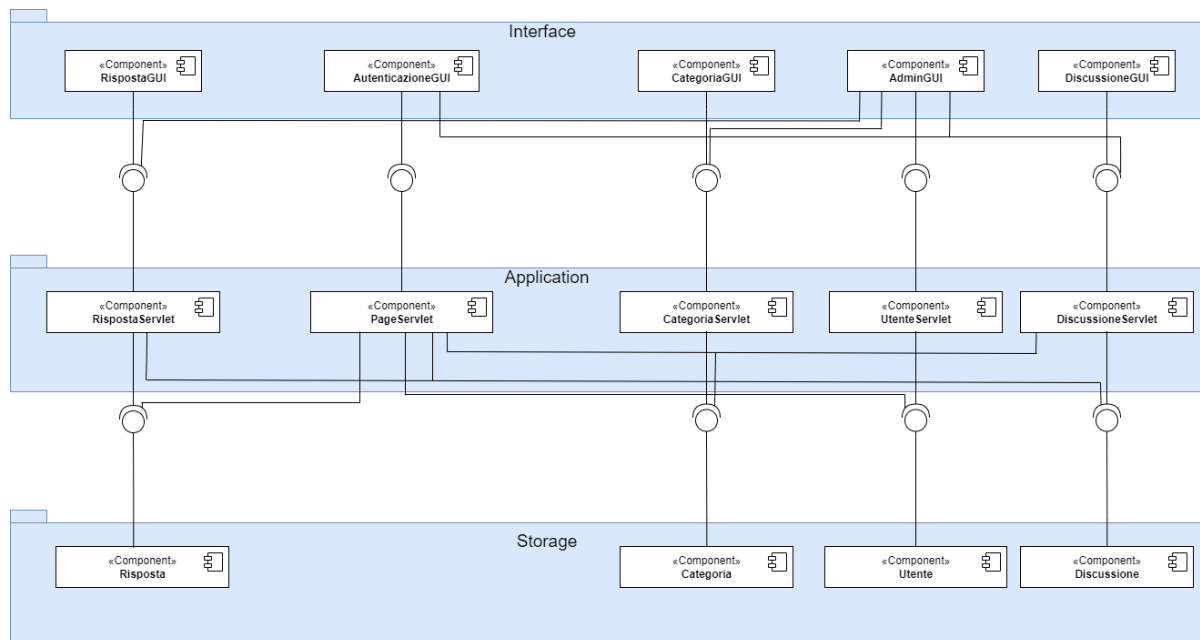
- Storage JPA verrà gestito da Connector JDBC.
- Persistenza sarà gestita attraverso un DBMS relazionale su sistema cloud VPS.

Di seguito una vista dettagliata di ciascun sottosistema evidenziando le componenti principale:

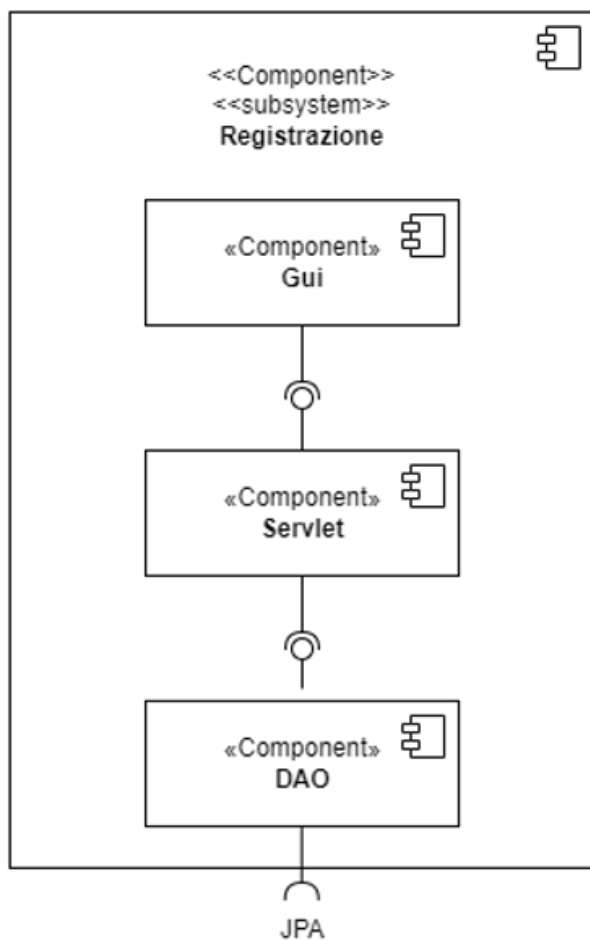
- **GUI:** Graphic User Interface, che contiene le varie view che saranno renderizzate per creare le pagine web da mostrare al cliente.
- **Controller:** si occupa della logica per il controllo del sistema.
- **Service:** si occupa della logica di business.
- **DAO:** Data Access Object, che si occupa di fornire accesso ai dati persistenti.



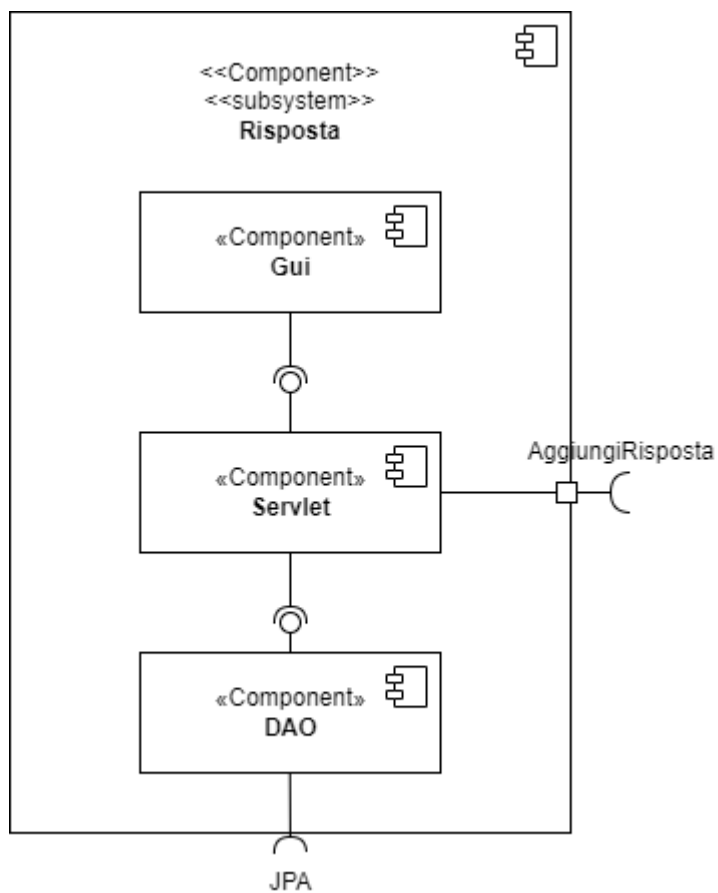
Schema Architetturale



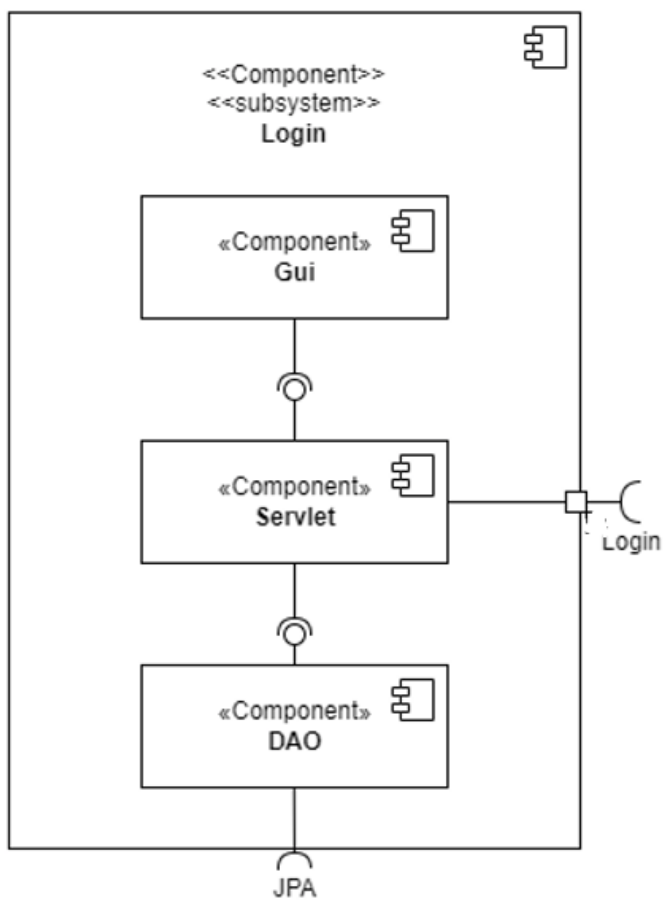
Sottosistema Registrazione



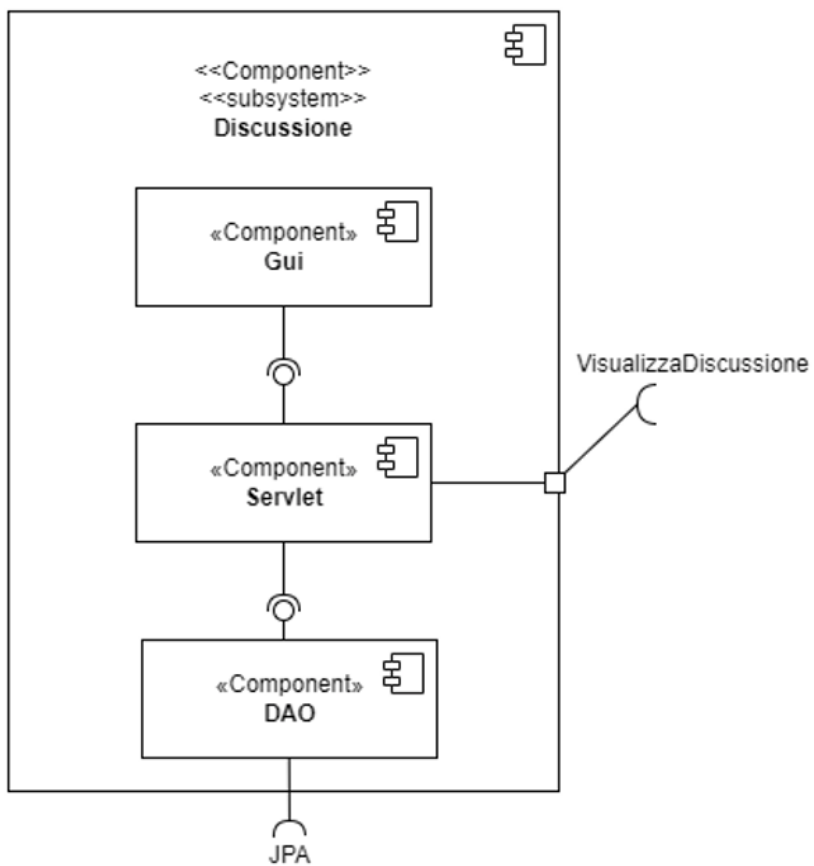
Sottosistema Risposta



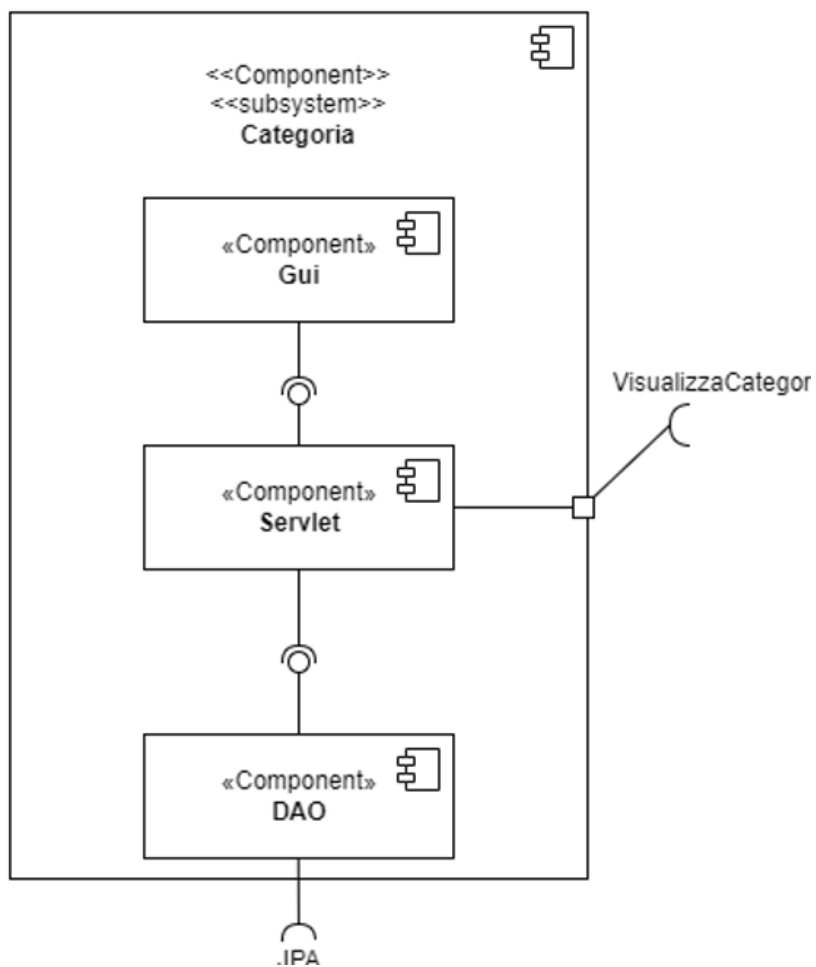
Sottosistema Login



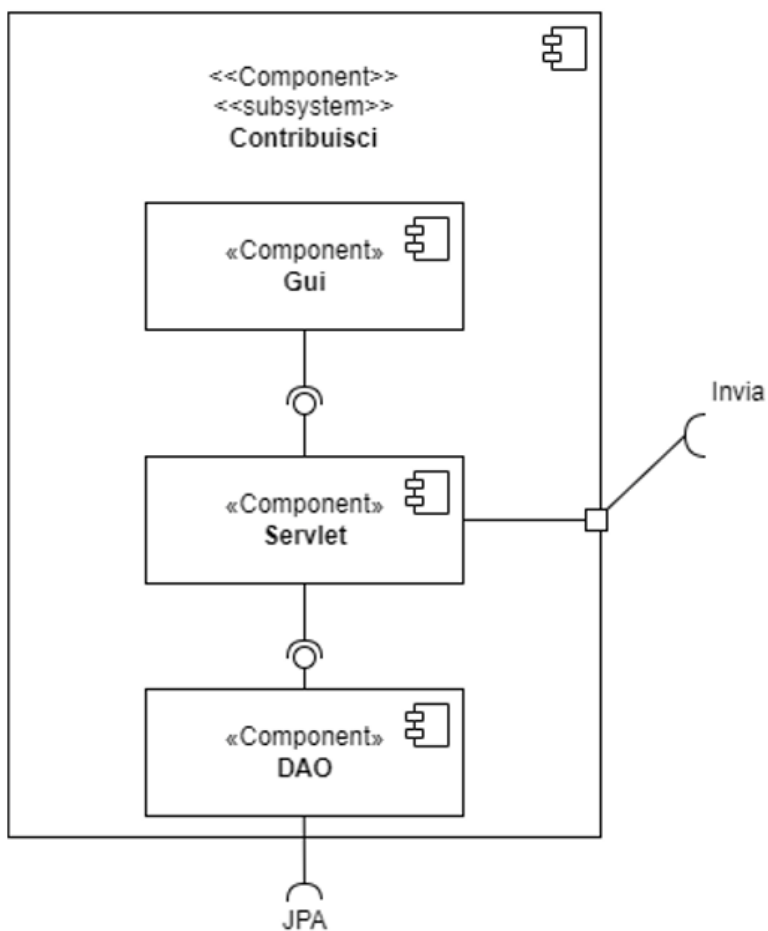
Sottosistema Discussione



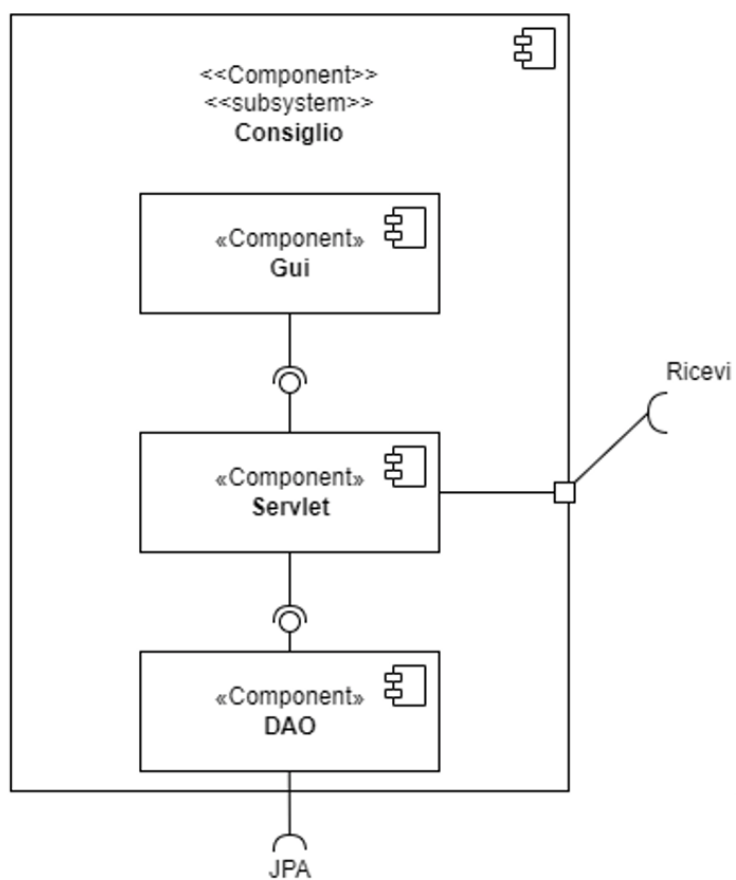
Sottosistema Categoria



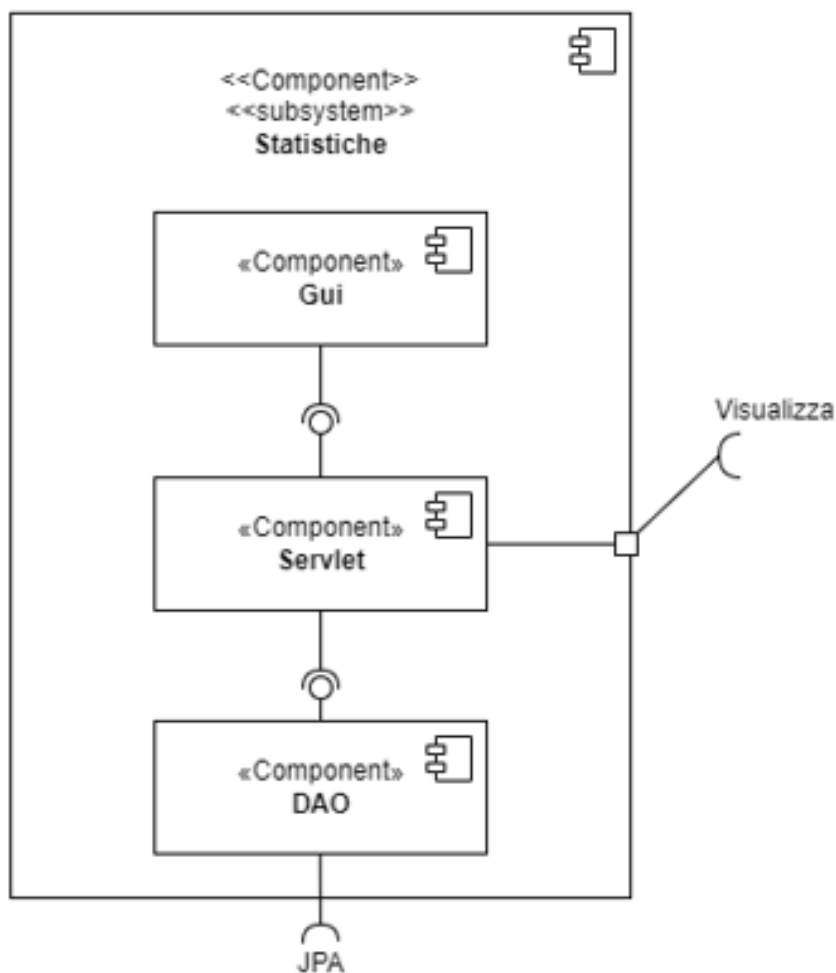
Sottosistema Contribuisci



Sottosistema consiglio



Sottosistema Statistiche



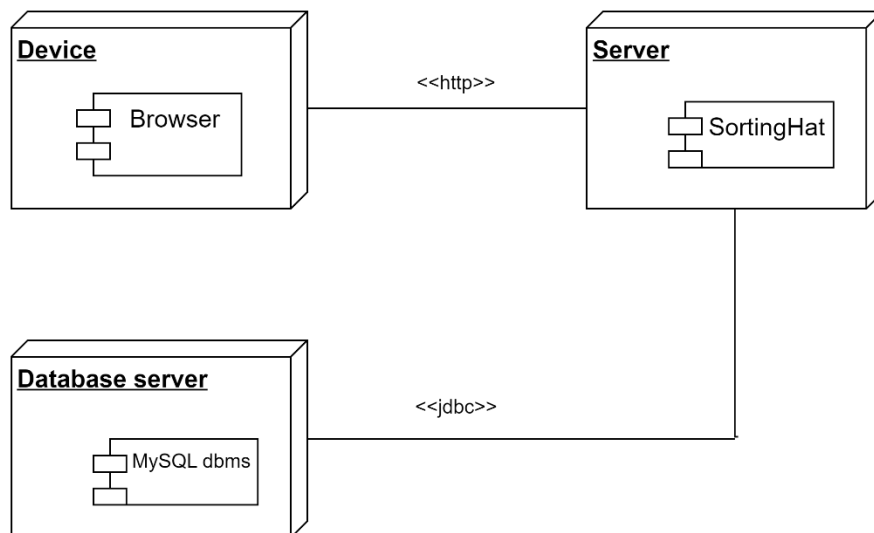
3.3. Hardware/software mapping

Il sistema essendo un applicazione web usufruirà di una struttura hardware composta da un server per interagire con le richieste dei client.

Consideriamo client qualsiasi terminale attraverso cui un utente, rispettando i protocolli **HTTP**, riesce a collegarsi al server (Apache Tomcat).

Il server avrà il compito di gestire la logica di applicazione e la persistenza dei dati utilizzando come componente software un **DBMS** per comunicare con più client utilizzando coppie di richieste e risposte **HTTP**.

Infine, poiché l'accesso al sistema da parte del client avverrà solamente attraverso un browser web risulta chiaro che al client basti avere una connessione internet e un browser installato.



3.4. Gestione della Persistenza

Sorting Hat necessita di salvare in memoria alcuni dati, mentre per altri non è necessario.

Alcuni dati da salvare:

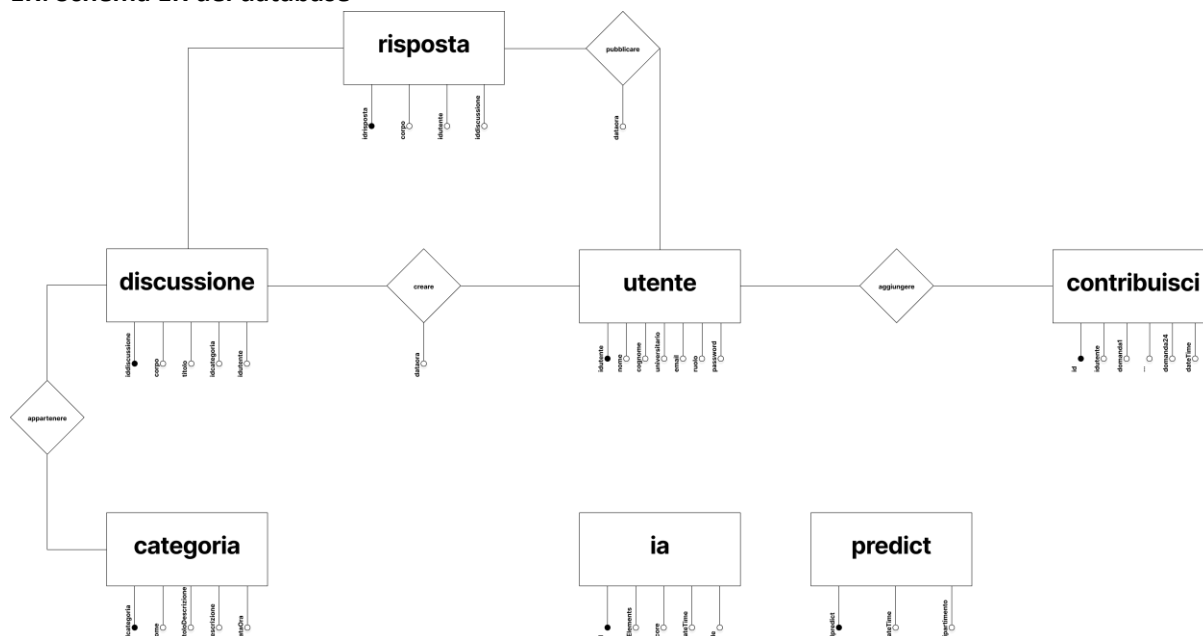
- Informazioni degli utenti registrati.
- Risposte ai form nella sezione “contribuisci”.
- Predict generati dal modulo di IA
- Model generati dal modulo di IA
- Thread relativi alle discussioni del

Alcuni dati da non salvare saranno quelli relativi alle risposte dei form della sezione “consiglio” poiché non verranno utilizzati se non nell’immediato output all’**utente ospite**.

Per garantire la persistenza e l’interrogazione dei dati verrà utilizzato il connettore **JDBC** per un database relazionale.

Inoltre il nostro database sarà hostato online per permettere l’accesso da macchine in remoto.

ER: Schema ER del database



Dizionario dei dati:

- Utente*

NOME	TIPO	NULL	KEY
IDUTENTE	INT (Autoincrement)	NOT NULL	PRIMARY KEY
NOME	VARCHAR(50)	NOT NULL	
COGNOME	VARCHAR(50)	NOT NULL	
UNIVERSITARIO	BOOLEAN	NOT NULL	
EMAIL	VARCHAR(319)	NOT NULL	
RUOLO	INT	NOT NULL	
PASSWORD	TEXT	NOT NULL	

- Discussione*

NOME	TIPO	NULL	KEY
IDDISCUSSIONE	INT (Autoincrement)	NOT NULL	PRIMARY KEY
CORPO	TEXT	NOT NULL	
DATAORA	DATETIME	NOT NULL	
TITOLO	VARCHAR(50)	NOT NULL	
IDCATEGORIA	INT	NOT NULL	FOREIGN KEY
IDUTENTE	INT	NOT NULL	FOREIGN KEY



- *Risposta*

NOME	TIPO	NULL	KEY
IDRISPOSTA	INT (Autoincrement)	NOT NULL	PRIMARY KEY
CORPO	TEXT	NOT NULL	
DATAORA	DATETIME	NOT NULL	
IDUTENTE	INT	NOT NULL	FOREIGN KEY
IDDISCUSSIONE	INT	NOT NULL	FOREING KEY

- *Categoria*

NOME	TIPO	NULL	KEY
IDCATEGORIA	INT (Autoincrement)	NOT NULL	PRIMARY KEY
NOME	VARCHAR(20)	NOT NULL	
DESCRIZIONE	TEXT	NOT NULL	

- *Contribuisci*

NOME	TIPO	NULL	KEY
IDCONTRIBUISCI	INT (Autoincrement)	NOT NULL	PRIMARY KEY
IDUTENTE	INT	NOT NULL	FOREIGN KEY
DOMANDA1	INT	NOT NULL	
...	INT	NOT NULL	
DOMANDA24	INT	NOT NULL	
DATETIME	DATETIME	NOT NULL	

- *Ia*

NOME	TIPO	NULL	KEY
IDIA	INT (Autoincrement)	NOT NULL	PRIMARY KEY
NELEMENTS	INT	NOT NULL	
SCORE	INT	NOT NULL	
DATETIME	DATETIME	NOT NULL	
FILE	TEXT	NOT NULL	

- *Predict*

NOME	TIPO	NULL	KEY
IDPREDICT	INT (Autoincrement)	NOT NULL	PRIMARY KEY
DATETIME	DATETIME	NOT NULL	
DIPARTIMENTO	INT	NOT NULL	



3.5. Access control and security

In Sorting Hat vi sono attori con ruoli diversi che hanno privilegi diversi per eseguire le operazioni.

Utilizzando una matrice degli accessi possiamo chiarire le differenze tra di loro.

Ogni entry è composta da un attore e ad una classe e definisce le operazioni permesse da quell'attore sulle istanze di quella classe.

Sottosistema	Gestione		
Attore	FORM	DOMANDA/RISPOSTE	UTENTE
Studiante-ordinario	-Esegue form "consigli" -vede statistiche	-Genera domanda/risposta -Visualizza domande/risposte esistenti	-modifica info individuali
Studiante-universitario	-Esegue form "consigli" -esegue form "contribuisci" -vede statistiche	-Genera domanda/risposta -Visualizza domande/risposte esistenti	-modifica info individuali
Amministratore	-Modifica Form -vede statistiche	-Genera domanda/risposta -Elimina domanda/risposta -Visualizza risposta	-Eliminazione e Gestione Account utenti
Ospite	-esegue form "consigli" -vede statistiche	-visualizza domande/risposte esistenti	/



3.6. Global software control

Il sistema Sorting Hat è un sito web basato sull'interazione con l'utente attraverso sequenze di richieste e risposte HTTP.

L'utente effettua un comando, invocando la richiesta HTTP al server, creando un evento gestito da uno specifico event handler lato server, il quale indirizzerà la richiesta al sottosistema che gestisce la logica di controllo e che si occuperà di rivolgersi ai servizi per la logica applicativa.

Pertanto, la scelta del meccanismo di controllo del flusso è ricaduta sulla tipologia event-driven, in cui il flusso è guidato dagli eventi.



3.7. Boundary conditions

Inizializzazione (Startup)

Identificativo <i>UC_BC_1-Startup</i>		<i>Attività forum</i>	<i>Data</i>		<i>07/01/22</i>
			<i>Vers.</i>	<i>1.0</i>	
			<i>Autore</i>	Di Sarno D., Ascione J.Di Gregorio E.	
Descrizione		<i>Attività di avvio del sistema.</i>			
Attore Principale		Amministratore			
Attori secondari		NA			
Entry Condition		L’amministratore effettua l’accesso al server			
Exit condition On success		Il sistema viene avviato con successo.			
Exit condition On failure		Avvio del sistema non riuscito.			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO					
1	Amministratore	Inserisce il comando per lo Startup del Sistema			
2	Sistema	controlla l’interezza dei dati caricati dal DB e notifica un eventuale errore			
Scenario/Flusso di eventi di ERRORE: Il sistema notifica un errore dei dati persistenti					
2.1	Sistema	Visualizza un messaggio di errore all’amministratore e termina l’operazione di Startup con insuccesso			
2.2	Amministratore	Corregge i dati corrotti			
2.3	Amministratore	Ripete l’avvio dal passaggio 1			



Terminazione(Shutdown)

Identificativo <i>UC_BC_1-Shutdown</i>		<i>Attività forum</i>	<i>Data</i>		<i>07/01/22</i>
			<i>Vers.</i>	<i>1.0</i>	
			<i>Autore</i>	Di Sarno D., Ascione J Di Gregorio E.	
Descrizione		<i>Attività di spegnimento del sistema.</i>			
Attore Principale		Amministratore			
Attori secondari		NA			
Entry Condition		L'amministratore effettua l'accesso al server AND Il Sistema è stato avviato AND Il sistema non è ancora spento			
Exit condition On success		Il sistema viene spento con successo.			
Exit condition On failure		Spegnimento del sistema non riuscito.			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO					
1	Amministratore	Inserisce il comando per lo Shutdown del Sistema			
2	Sistema	controlla la presenza di connessione interne/esterne ancora attive, se esse sono presenti il sistema le termina. Se non sono presenti connessioni viene terminata l'esecuzione del sistema.			
Scenario/Flusso di eventi di ERRORE: Il sistema nota connessioni aperte					
2.1	Sistema	Visualizza un messaggio di notifica all'amministratore e sospende l'operazione di terminazione con insuccesso			
2.2	Sistema	Blocca la generazione di nuove connessioni mentre termina quelle aperte.			
2.3	Sistema	Ripete il punto 2			
2.4	Sistema	Notifica lo spegnimento all'Amministratore			



Fallimento(Failure)

- Fallimenti legati ai dati persistenti

Identificativo <i>UC_BC_1-Dati corrotti o non accedibili</i>		<i>Attività forum</i>	<i>Data</i>		<i>07/01/22</i>
			<i>Vers.</i>	<i>1.0</i>	
			<i>Autore</i>	Di Sarno D., Ascione J.,Di Gregorio E	
Descrizione		<i>Attività in caso di errore dei dati persistenti o del loro accesso da parte del sistema.</i>			
Attore Principale		Amministratore			
Attori secondari		NA			
Entry Condition		Il Sistema non accede ai dati salvati nel DB OR I dati salvati sono corrotti			
Exit condition On success		Il Sistema riprende il funzionamento con successo.			
Exit condition On failure		Ripresa del funzionamento del Sistema non riuscito.			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO					
1	Sistema	Notifica il mancato accesso ai dati persistenti.			
2	Sistema	Sospende le connessioni inviando notifiche di errore.			
3	Amministratore	Esegue l'operazione di terminazione come esposto nell' UC_BC_2 .			
4	Amministratore	Ripristina i dati persistenti o il loro accesso.			
5	Amministratore	Esegue l'operazione di avvio come esposto nell' UC_BC_1 .			



- Fallimenti sistema

Identificativo <i>UC_BC_1-Fallimento del Sistema</i>		<i>Attività forum</i>	<i>Data</i>		<i>07/01/22</i>
			<i>Vers.</i>	<i>1.0</i>	
			<i>Autore</i>	Di Sarno D., Ascione J. „Di Gregorio E	
Descrizione		<i>Attività in caso di fallimento del sistema.</i>			
Attore Principale		Amministratore			
Attori secondari		NA			
Entry Condition		Il sistema termina in modo inaspettato			
Exit condition On success		Il Sistema riprende il funzionamento con successo.			
Exit condition On failure		Ripresa del funzionamento del Sistema non riuscito.			
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO					
1	Amministratore	Esegue il riavvio come esposto nell' UC_BC_1			



4. Subsystems services

Qui sono mostrati i sottoservizi dei sottosistemi.

Registrazione

SERVIZIO	DESCRIZIONE	INTERFACCIA
Registrazione Utente Universitario	Questa funzionalità permette di registrarsi sulla piattaforma come Universitario	AutenticazioneGUI
Registrazione Utente Ordinario	Questa funzionalità permette di registrarsi sulla piattaforma come Utente Ordinario	AutenticazioneGUI

Autenticazione

SERVIZIO	DESCRIZIONE	INTERFACCIA
Login Utente	Questa funzionalità permette di entrare sulla piattaforma come Utente	AutenticazioneGUI
Login Admin	Questa funzionalità permette di entrare sulla piattaforma come Utente	AutenticazioneGUI
Logout	Questa funzionalità permette di uscire dal proprio profilo	AutenticazioneGUI

Contribuisci

SERVIZIO	DESCRIZIONE	INTERFACCIA
Effettua Contribuisci	Questa funzionalità permette di compilare il form "Contribuisci"	AutenticazioneGUI

Consiglio

SERVIZIO	DESCRIZIONE	INTERFACCIA
Effettua consiglio	Questa funzionalità permette di compilare il form "Consiglio"	AutenticazioneGUI

Statistiche

SERVIZIO	DESCRIZIONE	INTERFACCIA
Visualizzo Statistiche	Questa funzionalità permette di visualizzare la pagina delle Statistiche	AutenticazioneGUI



Discussione

SERVIZIO	DESCRIZIONE	INTERFACCIA
Visualizza Discussioni	Questa funzionalità permette di visualizzare tutte le Discussioni	DiscussioneGUI
Crea Discussione	Questa funzionalità permette di creare una Discussione	DiscussioneGUI
Modifica Discussione	Questa funzionalità permette di modificare una Discussione	DiscussioneGUI
Elimina Discussione	Questa funzionalità permette di eliminare una Discussione	AdminGUI

Risposte

SERVIZIO	DESCRIZIONE	INTERFACCIA
Crea Risposta	Questa funzionalità permette Di creare una Risposta	RisposteGUI
Visualizza Risposte	Questa funzionalità permette di visualizzare tutte le Discussioni	RisposteGUI
Modifica Risposta	Questa funzionalità permette di modificare una Discussione	RisposteGUI
Elimina Risposta	Questa funzionalità permette di eliminare una Discussione	AdminGUI

Categoria

SERVIZIO	DESCRIZIONE	INTERFACCIA
Crea Categoria	Questa funzionalità permette Di creare una Risposta	CategoriaGUI
Modifica Categoria	Questa funzionalità permette di modificare una Discussione	CategoriaGUI
Elimina Categoria	Questa funzionalità permette di eliminare una Discussione	AdminGUI
Visualizza Categorie	Questa funzionalità permette di visualizzare tutte le Discussioni	CategoriaGUI