

# System Design Document Shareboard

Riferimento	SDD_G07
Versione	2.0
Data	15/03/2022
Destinatario	Studenti di Ingegneria del Software 2021/22
Presentato da	Antonio Romano (AR), Alessandro Saverio De Maio (AM), Carmine Leo (CL)

## Revision History

Data	Versione	Descrizione	Autori
11/11/2021	0.1	<ul style="list-style-type: none"><li>Introduzione</li></ul>	AR, AM, CL
12/11/2021	0.2	<ul style="list-style-type: none"><li>Architettura del software corrente</li></ul>	AR, AM, CL
03/01/2022	0.3	<ul style="list-style-type: none"><li>Panoramica del sistema proposto.</li></ul>	AR, AM, CL
04/01/2022	0.4	<ul style="list-style-type: none"><li>Decomposizione del sistema.</li></ul>	Antonio Romano
04/01/2022	0.5	<ul style="list-style-type: none"><li>Mapping HW/SW.</li><li>Gestione dei dati persistenti.</li></ul>	AR, AM, CL
07/01/2022	0.6	<ul style="list-style-type: none"><li>Tabelle delle entità</li></ul>	Carmine Leo
11/01/2022	0.7	<ul style="list-style-type: none"><li>Sicurezza e controllo degli accessi.</li><li>Controllo del software globale.</li></ul>	Alessandro Saverio De Maio
15/01/2022	0.8	<ul style="list-style-type: none"><li>Condizioni limite.</li><li>Servizi dei sottosistemi.</li></ul>	Carmine Leo
03/02/2022	0.9	<ul style="list-style-type: none"><li>Glossario.</li></ul>	AR, AM, CL
07/02/2022	0.10	<ul style="list-style-type: none"><li>Revisione schema concettuale.</li></ul>	Antonio Romano
19/02/2022	1.0	<ul style="list-style-type: none"><li>Revisione generale.</li></ul>	AR, AM
13/03/2022	2.0	<ul style="list-style-type: none"><li>Revisitazione decomposizione</li><li>Revisione completa</li></ul>	Antonio Romano

## Contenuti

Revision History .....	2
Introduzione.....	4
Scopo del sistema .....	4
Obiettivi di design .....	4
Design Trade-Offs .....	5
Definizioni acronimi e abbreviazioni .....	5
Riferimenti.....	6
Panoramica .....	6
Architettura del software proposto .....	6
Panoramica .....	6
Introduzione.....	6
Decomposizione del sistema .....	7
Mapping Hardware/Software.....	8
Gestioni dei dati persistenti.....	9
Schema logico .....	10
Sicurezza e controllo degli accessi.....	13
Controllo del software globale .....	14
Condizioni limite .....	14
Avvio del sistema .....	14
Spegnimento del sistema.....	15
Fallimento del sistema .....	15
Servizi dei sottosistemi.....	16
Glossario .....	18

## Introduzione

---

### Scopo del sistema

Il sistema che si vuole realizzare ha 2 obiettivi: facilitare lo scambio di informazioni tra persone che condividono interessi simili e permettere agli utenti di scoprire nuove passioni.

Il sistema è una piattaforma online dove gli utenti possono pubblicare contenuti che ritengono essere di interesse per la community e rispondere ai contenuti creati dagli altri utenti.

### Obiettivi di design

Gli obiettivi di design identificati per il nostro sistema sono i seguenti:

#### Criteri di performance

- **Tempo di risposta:**  
Il sistema deve essere in grado di eseguire le proprie operazioni ed inviare la risposta in un tempo inferiore o uguale ai 2 secondi.
- **Troughput:**  
Il sistema dovrà essere in grado di soddisfare un minimo di 1000 richieste simultanee.
- **Spazio:**  
La memoria necessaria al sistema dipenderà dalla memoria richiesta dal database per il mantenimento dei dati persistenti e dalle immagini memorizzate nel file system

#### Criteri di affidabilità

- **Robustezza:**  
Il sistema effettua una verifica dei dati in input sia lato client che lato server, informando l'utente di eventuali input errati con messaggi di errore quando la verifica avviene lato client, nel caso l'utente aggirasse le verifiche lato client, il sistema mostrerà un messaggio di errore in una pagina dedicata dando la possibilità all'utente di ritornare alla homepage.
- **Disponibilità:**  
Il sistema deve essere attivo 24/7 con la possibilità di andare offline per manutenzione nelle ore notturne per un massimo di 2 ore.
- **Tolleranza ai guasti:**  
Al fine di minimizzare possibili perdite di dati, una volta alla settimana sarà effettuato il backup del database da parte di un admin.
- **Sicurezza:**  
L'accesso al sistema sarà effettuato tramite e-mail e password. Per aumentare la sicurezza le password saranno criptate tramite l'utilizzo dell'algoritmo PBKDF2WithHmacSHA1 con un apposito salt generato casualmente per ogni utente.

#### Criteri di costi

- **Costi di sviluppo:**  
È stimato un costo di sviluppo e di progettazione di 150 ore (50 ore per ogni membro del progetto).

#### Criteri di manutenzione

- **Estendibilità:**  
È possibile aggiungere nuove funzionalità tramite l'implementazione di nuove classi che andranno opportunamente aggiunte ai sottosistemi presenti o la creazione di nuovi sottosistemi.
- **Modificabilità:**  
Il sistema è pensato in modo da essere facilmente modificabile attraverso una strutturazione del progetto in componenti facilmente rimpiazzabili
- **Adattabilità:**  
Il sistema è ampiamente adattabile, trattandosi di un sistema i cui contenuti spaziano dalle notizie ai contenuti informativi fino a contenuti di intrattenimento pubblicati dagli utenti, possiamo adattare il sistema a diversi domini applicativi.

- **Portabilità:**

Il sistema è per sua natura portabile su ogni genere di dispositivo che disponga di un browser installato, essendo una web application responsive, è pensata proprio per facilitarne l'utilizzo anche da smartphone via browser.

- **Leggibilità del codice:**

Il codice del sistema adotterà le più comuni pratiche di programmazione e di stile, inserendo - dove necessario linee - di commento che hanno lo scopo di commentare la funzionalità e il significato dell'algoritmo, in questo modo si faciliterà la lettura ad ogni membro del team attuale o futuro che dovrà effettuare interventi di manutenzione.

- **Tracciabilità dei requisiti:**

Si terrà traccia della struttura, delle relazioni e delle dipendenze tra i requisiti e gli altri work product al fine di identificare le componenti coinvolte ad una modifica nel sistema e di conseguenza a garantire una maggiore velocità di risposta ai cambi di requisiti

## Criteri di usabilità

- **Utilità:**

Il sistema mira ad espandere gli interessi degli utenti e ad incentivare la condivisione di notizie e contenuti come gli how-to-content che mirano a spiegare come fare una determinata cosa, in questo modo un utente oltre che a socializzare con altri utenti con interessi simili, può anche apprendere nuove nozioni su argomenti che mai avrebbe pensato di trovare.

- **Usabilità:**

Il sistema sarà di facile utilizzo grazie ad un interfaccia User-Friendly e minimale che permetterà un utilizzo confortevole anche da smartphone. L'utente deve essere in grado di imparare ad usare la piattaforma durante l'utilizzo.

## Design Trade-Offs

- **Spazio vs Velocità:**

Il sistema darà priorità alla velocità per fornire una buona esperienza utente a discapito della memoria, per questo motivo si utilizzeranno meccanismi di caching e di ridondanza.

- **Tempo di rilascio vs Funzionalità:**

Se i tempi di rilascio di tutte le funzionalità non possono essere rispettati, saranno rilasciate soltanto le funzionalità che hanno la massima priorità in modo da rientrare nei tempi previsti.

- **Tempo di rilascio vs Qualità:**

Per rispettare i tempi di rilascio in caso di ritardi, si concentreranno gli sforzi per ridurre il numero di bug nelle funzionalità ad alta priorità, successivamente al rilascio saranno sistemati tutti i restanti difetti del codice.

## Definizioni acronimi e abbreviazioni

Di seguito una lista di definizioni, acronimi e abbreviazioni usati nel documento:

- **Hibernate:** Framework per la gestione dei dati persistenti.
- **DBMS:** DataBase Management System
- **RDBMS:** Relational DataBase Management System
- **FS:** File System
- **DB:** Database
- **API:** Application Programming Interface
- **Java EE:** Java Enterprise Edition
- **JSP:** Java Server Pages
- **SQL:** Structured Query Language
- **JPA:** Java Persistence API
- **CDI:** Context (and) Dependency Injection

## Riferimenti

### Documenti

- Requirements Analysis Document
- Object Design Document
- Test Plan
- Matrice di tracciabilità
- Manuale di installazione
- Manuale utente

### Bibliografia

Bruegge, B., & Dutoit, A. H. (2014). *Object-oriented software engineering*. Pearson.

Sommerville, I. (2019). *Software engineering, tenth edition*. Pearson Education.

## Panoramica

Il presente documento è composto dalle seguenti sezioni:

1. **Introduzione:** Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema propone di raggiungere.
2. **Architettura software proposta:** una descrizione come il sistema sarà definito e partizionato in sottosistema, il loro mapping Hardware/Software, la gestione dei dati persistenti. Infine una presentazione della struttura dei sottosistemi e le condizioni limite riguardanti il sistema.
3. **Servizi dei sottosistemi:** descrive i servizi forniti da ogni sottosistema
4. **Glossario:** contiene la lista dei termini usati nel documento.

## Architettura del software proposto

### Panoramica

La sezione attuale contiene una serie di sottosezioni, di seguito una breve descrizione.

1. **Introduzione:** fornisce una visione preliminare delle scelte architetturali adottate per la realizzazione del sistema
2. **Decomposizione del sistema:** descrive la decomposizione in sottosistemi e le responsabilità di ciascuno di essi
3. **Mapping hardware/ software:** descrive l'assegnazione dei sottosistemi a nodi hardware e componenti off-the-shelf
4. **Gestione dei dati persistenti:** descrive i dati persistenti memorizzati dal sistema e l'infrastruttura richiesta a tal scopo
5. **Sicurezza e controllo degli accessi:** definisce i servizi del sistema accessibili per ogni attore
6. **Controllo del software globale:** descrive il flusso di controllo globale del software
7. **Condizioni limite**

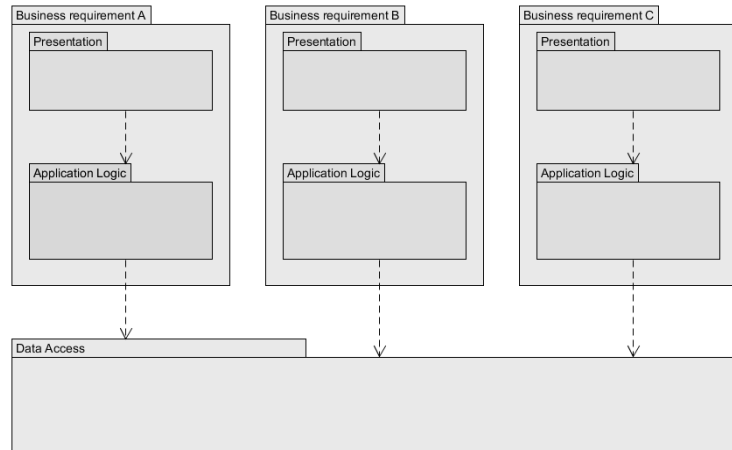
### Introduzione

Il sistema proposto segue un'architettura chiusa. Il sistema è suddiviso in 3 layer che ricoprono differenti responsabilità di carattere tecnico:

- **Interface Layer:** contiene la logica di presentazione. Rappresenta i confini del sistema
  - Tale layer è composto dagli oggetti Boundary identificati nel Requirements Analysis Document
- **Application Logic Layer:** contiene la logica applicativa
  - Tale layer è composto dagli oggetti Control identificati nel Requirements Analysis Document

- **Data Access Layer:** contiene le entità identificate nel Requirements Analysis Document e la logica per accedervi

Gli strati di Application Logic e di Interface sono a loro volta partizionati in base ai requisiti di business: questo consente una maggior velocità di risposta a eventuali cambi di requisiti dovuta a una maggior centralizzazione delle modifiche dovute.



I vantaggi di un'architettura chiusa stratificata sono i seguenti:

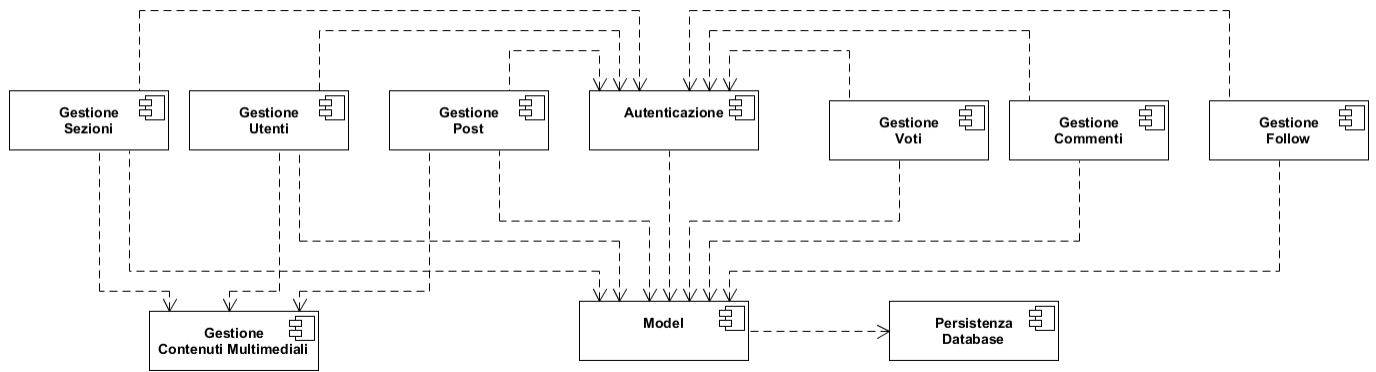
- La separazione di responsabilità tecniche consente a differenti team di sviluppatori con specifici set di competenze di lavorare a differenti layer del sistema
  - La comunicazione tra due layer avviene mediante apposite API interne definite
- Risposta veloce ad un cambio di tecnologie per la logica di presentazione o la logica di persistenza
- Disaccoppiamento tra logica di presentazione e logica di persistenza
- Facilità di testing: un'architettura stratificata permette una facile modalità di testing seguendo l'approccio bottom-up

## Decomposizione del sistema

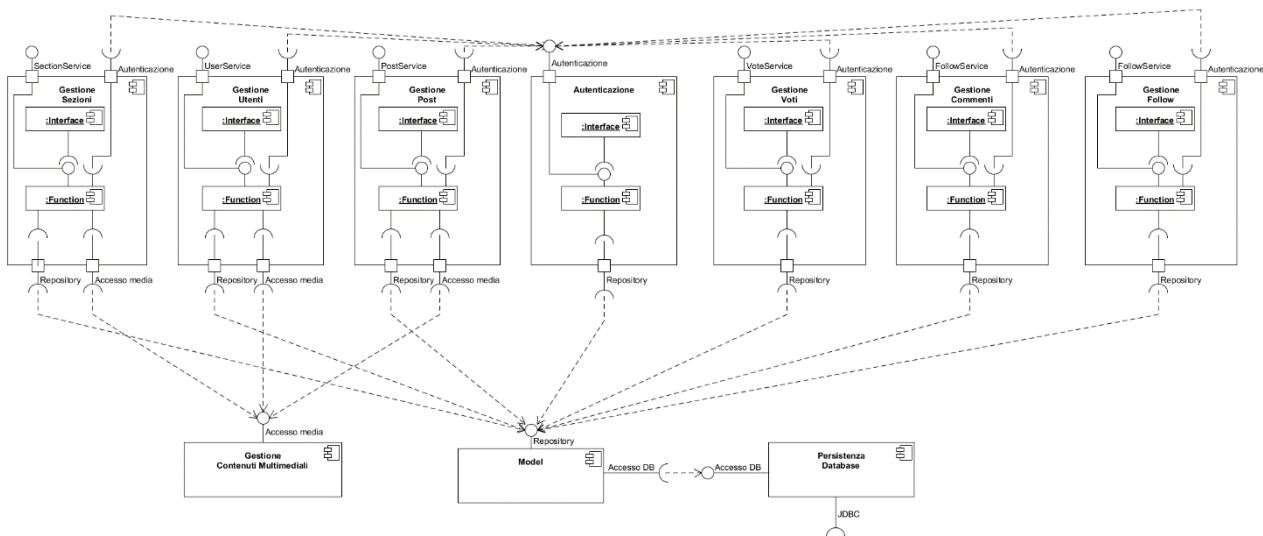
Di seguito una lista dei sottosistemi individuati:

- Autenticazione
- Gestione follow
- Gestione voti
- Gestione commenti
- Gestione utenti
- Gestione sezioni
- Gestione post
- Gestione contenuti multimediali
- Model
- Persistenza Database

Il seguente component diagram evidenzia le dipendenze tra i sottosistemi individuati:

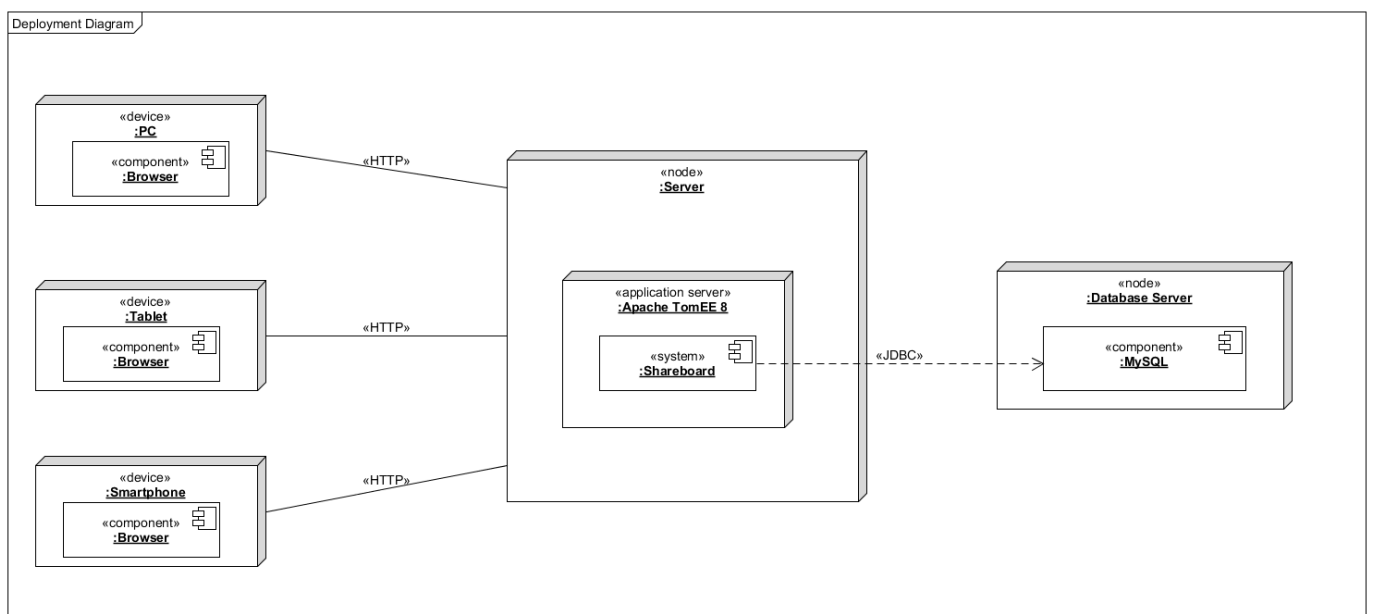


Il seguente component diagram costituisce una raffinazione del diagramma precedente, evidenziando la realizzazione dei sottosistemi e i servizi che questi offrono.



I servizi offerti dai sottosistemi sono definiti in maggior dettaglio nella sezione Servizi dei sottosistemi di questo documento

## Mapping Hardware/Software



Per la realizzazione del sistema Shareboard, abbiamo scelto un framework basato su Java EE 8.



Java EE (Java Enterprise Edition) è un insieme di standard le cui implementazioni sono tipicamente destinate allo sviluppo di applicativi Enterprise e Web-based. Questo consente a diversi provider di fornire una propria implementazione conforme agli standard imposti da Java EE, garantendo quindi maggiore facilità di migrazione tra diversi provider.

Il provider Java EE scelto per lo sviluppo di Shareboard è Apache TomEE 8. I seguenti componenti sono inclusi in TomEE e utilizzati a loro volta nello sviluppo del sistema:

- Apache Tomcat: Server HTTP e Servlet container per il processing di richieste e risposte HTTP
  - Fornisce supporto alle JSP, per la generazione di contenuti dinamici in formato HTML
- Apache OpenWebBeans: Java CDI container che fornisce funzionalità come iniezione contestuale di dipendenze, interceptor, listener, bean life-cycle management

Per la persistenza dei dati è stata adottata la seguente strategia:

- Memorizzazione delle immagini nel file system
- Memorizzazione delle entità persistenti in un database relazionale usando MySQL, tenendo traccia anche dei percorsi delle immagini nel file system

La comunicazione del sistema con il database relazionale è attuata per mezzo dei driver JDBC appositi per il DBMS di riferimento.

Per il mapping tra oggetti e schema del database è stato adottato il framework JBoss Hibernate 5.6, parte dello standard JPA di Java EE. Le strategie per la gestione dei dati persistenti sono definite in maggior dettaglio nella sezione Gestioni dei dati persistenti di questo documento.

Come supporto alla stesura del codice di scripting client-side in JavaScript, è stata adottata la libreria JQuery.

## Gestioni dei dati persistenti

Per la gestione dei dati persistenti si utilizzerà un DBMS Relazionale poiché offrono molteplici vantaggi, tra cui la scalabilità, accesso efficiente ai dati, minimizzazione di incoerenze e duplicazioni, inoltre permette un accesso concorrente ai dati. Il RDBMS scelto è MySQL.

Per la memorizzazione delle immagini viene fatto uso del file system situato nello stesso nodo in cui è situato l'application server. Il DBMS tiene traccia quindi del percorso relativo delle immagini nel file system menzionato.

## Schema logico

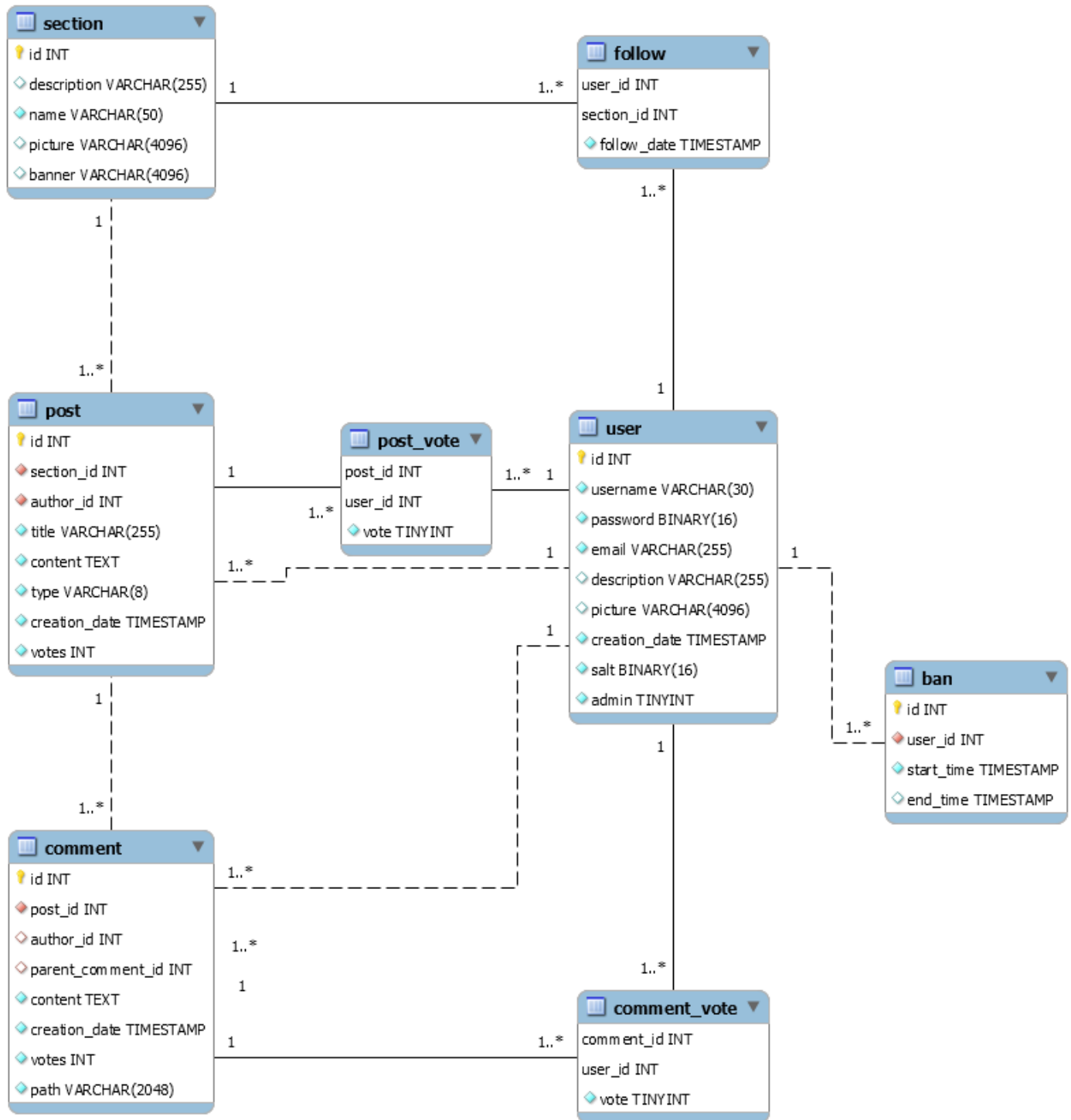


Tabella User

Nome	Tipo	Constraints	Key
ID	INT	NOT NULL, AUTO INCREMENT, UNIQUE	PRIMARY KEY
USERNAME	VARCHAR(30)	NOT NULL, UNIQUE	

PASSWORD	BINARY(16)	NOT NULL	
EMAIL	VARCHAR(255)	NOT NULL, UNIQUE	
DESCRIPTION	VARCHAR(255)	DEFAULT NULL	
PICTURE	VARCHAR(4096 )	DEFAULT NULL	
CREATION_DATE	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	
SALT	BINARY(16)	NOT NULL	
ADMIN	TINYINT	NOT NULL	

Tabella Ban

Nome	Tipo	Constraints	Key
ID	INT	NOT NULL, AUTO INCREMENT	PRIMARY KEY
USER_ID	INT	NOT NULL	FOREIGN KEY
START_TIME	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	
END_TIME	TIMESTAMP	DEFAULT NULL	

Tabella Comment

Nome	Tipo	Constraints	Key
ID	INT	NOT NULL, AUTO INCREMENT	PRIMARY KEY
POST_ID	INT	NOT NULL	FOREIGN KEY
AUTHOR_ID	INT	NOT NULL	FOREIGN KEY
PARENT_COMMENT_ID	INT	DEFAULT NULL	FOREIGN KEY
CONTENT	TEXT	NOT NULL	
CREATION_DATE	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	
VOTES	INT	NOT NULL	
PATH	VARCHAR(2048)	NOT NULL	

Tabella Comment\_Vote

Nome	Tipo	Constraints	Key
COMMENT_ID	INT	NOT NULL	FOREIGN KEY
USER_ID	INT	NOT NULL	FOREIGN KEY
VOTE	TINYINT	NOT NULL	

Tabella Follow

Nome	Tipo	Constraints	Key
USER_ID	INT	NOT NULL	FOREIGN KEY
SECTION_ID	INT	NOT NULL	FOREIGN KEY
FOLLOW_DATE	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	

Tabella Post

Nome	Tipo	Constraints	Key
ID	INT	NOT NULL, AUTO INCREMENT	PRIMARY KEY
SECTION_ID	INT	NOT NULL	FOREIGN KEY
AUTHOR_ID	INT	NOT NULL	FOREIGN KEY
TITLE	VARCHAR(255)	NOT NULL	
CONTENT	TEXT	NOT NULL	
TYPE	VARCHAR(8)	NOT NULL, DEFAULT 'TEXT'	
CREATION_DATE	TIMESTAMP	NOT NULL, DEFAULT CURRENT_TIMESTAMP	
VOTES	INT	NOT NULL	

Tabella Post\_Vote

Nome	Tipo	Constraints	Key
POST_ID	INT	NOT NULL	FOREIGN KEY
USER_ID	INT	NOT NULL	FOREIGN KEY
VOTE	TINYINT	NOT NULL	

Tabella Section

Nome	Tipo	Constraints	Key
ID	INT	NOT NULL, AUTO INCREMENT	PRIMARY KEY
DESCRIPTION	VARCHAR(255)	DEFAULT NULL	
NAME	VARCHAR(50)	NOT NULL, UNIQUE	
PICTURE	VARCHAR(4096)	DEFAULT NULL	
BANNER	VARCHAR(4096)	DEFAULT NULL	

### Gestione delle ridondanze

Sono mantenuti i seguenti dati derivabili nella memoria persistente:

- La colonna **Comment.Votes**
- La colonna **Comment.Path**
- La colonna **Post.Votes**

I valori degli attributi **Comment.Votes** e **Post.Votes** costituiscono rispettivamente la somma dei voti ad un commento e la somma dei voti ad un post. Tali attributi derivabili sono stati mantenuti nello schema del DB al fine di evitare di doverli calcolare ogni volta, a causa della frequente lettura di tali attributi (Vedi Requirements Analysis Document).

I valori della colonna **Comment.Path** costituiscono il percorso materializzato di un commento a partire dal nodo radice nella gerarchia dei commenti. Tale attributo è derivabile per far fronte alla necessità di visualizzare un commento insieme a tutti i suoi discendenti.

I valori derivabili sono aggiornati per mezzo di appositi trigger che mantengono gli attributi allineati ad ogni rimozione/cancellazione/aggiornamento di tuple correlate.

### Sicurezza e controllo degli accessi

Shareboard è un sistema in cui diversi attori possono eseguire diverse funzioni. La seguente tabella riporta i servizi a cui ciascun attore può accedere.

Oggetti/Attori	Admin	Utente	Utente non autenticato
Gestione Utente	Login	Login	Effettua Registrazione
Interazione	SottomettiPost	SottomettiPost	
	CommentaPost	CommentaPost	
	RispondiCommento	RispondiCommento	
	VotaCommento	VotaCommento	
	VotaPost	VotaPost	
	CancellaCommento Personale	CancellaCommento Personale	

	CancellaPostPersonale	CancellaPostPersonale	
Navigazione	VisualizzaFeed RicercaPost SeguiSezione AnnullaSeguiSezione VisualizzaSezione VisualizzaProfiloUtente	VisualizzaFeed RicercaPost SeguiSezione AnnullaSeguiSezione VisualizzaSezione VisualizzaProfiloUtente	VisualizzaFeed RicercaPost SeguiSezione AnnullaSeguiSezione VisualizzaSezione VisualizzaProfiloUtente
Moderazione	CancellaPost CancellaCommento Ban RimuoviBan		
Amministrazione	AggiungiSezione ModificaSezione CancellaSezione		

## Controllo del software globale

Il controllo software globale è **event-driven**. Le Java Servlet permettono di definire il comportamento sotto forma di metodo, eseguito dal Servlet Container alla ricezione di una richiesta HTTP. È possibile definire più metodi, eseguiti in base all'URL e al metodo della richiesta.

Per l'elaborazione di una richiesta, il Servlet Container crea un nuovo thread che avrà durata di vita pari all'intera durata dell'elaborazione della richiesta. Pertanto, nel flusso di controllo del software globale sono coinvolti più thread.

## Condizioni limite

Di seguito saranno presentate le boundary condition inerenti all'avvio del sistema, spegnimento del sistema e fallimento del sistema.

### Avvio del sistema

<b>Identificativo</b> UCBC_1	<b>Avvio del sistema</b>	Data	15/01/2022
		Vers.	0.00.001
		Autore	Carmine Leo
<b>Descrizione</b>	Lo UC permette l'avvio del sistema		
<b>Attore Principale</b>	Admin		
<b>Attori secondari</b>	NA		

<b>Entry Condition</b>		L'amministratore accede al server
<b>Exit Condition</b> On success		Il sistema viene avviato correttamente
<b>Exit Condition</b> On failure		Il sistema non viene avviato
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>		
1	Admin:	Esegue sulla macchina il comando che avvia il sistema
2	Sistema:	Verifica la connessione al database
3	Sistema:	Se il database non è stato inizializzato, il sistema inizializza il database creando gli opportuni schemi, trigger e aggiungendo gli opportuni record
4	Sistema:	Il sistema rende i servizi disponibili e rende disponibili le funzionalità agli utenti
<b>Scenario/Flusso di eventi Alternativo: I dati persistenti sono danneggiati</b>		
2a.1	Sistema:	Notifica all'amministratore la presenza di errori dei dati persistenti e non effettua l'avvio
2a.2	Admin	Corregge gli errori di persistenza e riesegue il punto 1
<b>Scenario/Flusso di eventi Alternativo: Connessione al database fallita</b>		
2b.1	Sistema:	Notifica all'amministratore l'impossibilità di stabilire una connessione con il database
2b.2	Admin	Individua le cause del fallimento e riesegue il punto 1

## Spegnimento del sistema

<b>Identificativo</b> UCBC_2		<b>Spegnimento del sistema</b>	Data	15/01/2022
			Vers.	0.00.001
			Autore	Carmine Leo
<b>Descrizione</b>		Lo UC permette lo spegnimento del sistema		
<b>Attore Principale</b>		<b>Admin</b>		
<b>Attori secondari</b>		NA		
<b>Entry Condition</b>		L'amministratore accede al server AND Il sistema è stato precedentemente avviato AND Il sistema è in esecuzione		
<b>Exit Condition</b> On success		Il sistema viene spento correttamente		
<b>Exit Condition</b> On failure		Il sistema non viene spento		
<b>FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO</b>				
1	Admin:	Invia un segnale di spegnimento al sistema		
2	Sistema:	Termina le connessioni con l'esterno, dealloca le risorse acquisite e arresta il sistema		

## Fallimento del sistema

<b>Identificativo</b> UCBC_3	<b>Fallimento del sistema</b>		Data	15/01/2022
			Vers.	0.00.001

		Autore	Carmine Leo
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 correttamente		
Exit Condition On failure	Il sistema non viene riavviato		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Admin:	Include UCBC_1	

## Servizi dei sottosistemi

Servizio	Descrizione	Interfaccia
Aggiunta di un ban	Questa funzionalità permette l'aggiunta di un ban ad un utente.	UserService
Rimozione di un ban	Questa funzionalità permette di rimuovere un ban.	UserService
Login	Questa funzionalità permette il login per un utente.	Autenticazione
Sign Up	Questa funzionalità permette la registrazione ad un utente.	UserService
Aggiunta di un commento	Questa funzionalità permette l'aggiunta di un commento.	CommentService
Rimozione di un commento	Questa funzionalità permette di rimuovere un commento.	CommentService
Modifica di un commento	Questa funzionalità permette di modificare un commento.	CommentService
Aggiunta di un follow	Questa funzionalità permette ad un utente di seguire una sezione.	FollowService
Rimozione di un follow	Questa funzionalità permette di non seguire ulteriormente una sezione.	FollowService
Visualizza immagine	Questa funzionalità permette di visualizzare un'immagine.	Accesso Media
Aggiunta di un post	Questa funzionalità permette di aggiungere un post.	PostService
Rimozione di un post	Questa funzionalità permette di rimuovere un post.	PostService
Visualizzazione di un post	Questa funzionalità permette di visualizzare un post.	PostService
Aggiunta di una sezione	Questa funzionalità permette di aggiungere una sezione.	SectionService
Rimozione di una sezione	Questa funzionalità permette di rimuovere una sezione.	SectionService
Visualizzazione di una sezione	Questa funzionalità permette di visualizzare una sezione.	SectionService



Rimozione di un utente	Questa funzionalità permette di rimuovere un utente.	UserService
Aggiunta di un voto ad un commento	Questa funzionalità permette di assegnare un voto ad un commento.	VoteService
Aggiunta di un voto ad un post	Questa funzionalità permette di assegnare un voto ad un post.	VoteService
Rimozione di un voto ad un commento	Questa funzionalità permette di rimuovere un voto ad un commento.	VoteService
Rimozione di un voto ad un post	Questa funzionalità permette di rimuovere un voto ad un post.	VoteService
CRUD Entità	Permette l'accesso a funzionalità di creazione, rimozione, modifica e lettura delle entità	Repository
Accesso database	Fornisce l'accesso ad operazioni del database relazionale	Accesso DB

## Glossario

Sigla/Termine	Definizione
Account	Rappresentazione dell'utente in formato digitale.
Admin	Amministratore della piattaforma
Ban	Divieto di interazione sulla piattaforma per un particolare utente
Commento	Messaggio di risposta ad un post o ad un altro commento
Feed	Insieme dei post appartenenti alle sezioni seguite da un particolare utente
Follow	Una istanza di relazione "segue" tra un utente e una sezione
Mock-up	Una rappresentazione visuale della User Interface, utile al committente per capire come il prodotto sarà fruibile all'utente.
Piattaforma	Base software o hardware su cui sono sviluppate o eseguite applicazioni.
Post	Contenuto condiviso da un utente in una determinata sezione, può contenere il titolo, il testo e anche un'immagine.
Sezione	Una categoria di argomenti creata da un amministratore, dove gli utenti possono condividere post coerenti in termini contenutistici alla categoria.