Laurea in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof. C. Gravino

System Design
Progetto
GuardaTV

Riferimento	
Versione	1.3
Data	13/02/2022
Destinatario	Studenti di Ingegneria del Software 2021/22
Presentato da	Gruppo 16: N. Cacace S. Pastore A. Prezioso A. Ricchetti
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
12/11/2021	0.1	Design Goal	N. Cacace S. Pastore A. Prezioso A. Ricchetti
19/11/2021	0.2	Identify subsystem e mapping hardware-software	N. Cacace S. Pastore A. Prezioso A. Ricchetti
26/11/2021	0.3	Dati persistenti, controllo accessi, boundary condition	N. Cacace S. Pastore A. Prezioso A. Ricchetti
1/12/2021	0.4	Global software control e modifiche minori	N. Cacace S. Pastore A. Prezioso A. Ricchetti
3/12/2021	0.5	Descrizioni e revisione	N. Cacace S. Pastore A. Prezioso A. Ricchetti
11/12/2021	0.51	Indice e modifiche minori	S. Pastore
12/12/2021	1.0	Modifica Trade-off e Revisione	N. Cacace S. Pastore A. Prezioso
20/01/2022	1.1	Descrizione access control and security	N. Cacace
12/02/2022	1.2	Modifica Persistent data management ed aggiunta Class diagram	N.Cacace
13/02/2022	1.3	Modifica Subject decomposition ed Access Control and security	A.Prezioso

Sommario

1. Introduzione 4
1.1. Scopo del sistema 4
1.2. Design Goal 4
1.2.1. Trade-off 5
1.3. Definizioni, acronimi e abbreviazioni 5
1.4. Riferimenti 6
1.5. Panoramica 6
2. Architettura Software 6
2.1. Overview 6
2.2. Subsystem decomposition 6
2.3. Hardware/Software mapping 8
2.4. Persistent data managment 9
2.4.1. Identifiyng persistent object 9
2.4.2. Storage strategy 11
2.5. Access control and security 12
2.6. Global software control 13
2.7. Boundary Condition 13
2.7.1. Avvio del sistema 13
2.7.2. Shut Down 13
2.7.3. Fallimento 14
2.7.4. Use Cases 14
3. Subsystem Services 15

1. Introduzione

1.1. Scopo del sistema

Il sistema GuardaTV è stato creato con lo scopo di aiutare gli utenti del sito riguardo la scelta di un film o una serie TV da guardare. Infatti il sistema comprende un sistema di recensioni grazie al quale l'utente può informarsi riguardo un contenuto da usufruire. Inoltre GuardaTV permette la gestione di liste personalizzate.

1.2. Design Goal

- **Performance:**
  - **Tempi di Risposta:** Il sistema deve garantire tempi di risposta nell'ordine della decina di secondi;
- **Dependability**
  - **Robustness:** Tutti i campi vengono verificati sia lato client che server in modo da sopportare input errati o non validi
  - **Security:** GuardaTV non deve permettere accesso non autorizzato ai dati degli utenti;
- **Cost**
  - **Development Cost:** Il tempo per lo sviluppo di GuardaTV non deve superare le 50h/persona.
- **Maintenance**
  - **Estensibilità:** Possibilità di aggiungere nuove tipologie di contenuti, filtri di ricerca e di ordinamento.
  - **Modificabilità:** Garantire la leggibilità del codice in modo da rendere più semplice ed agevole la modifica delle componenti.
- **End user criteria**
  - **Usabilità:** Le funzioni di gestione delle liste devono essere intuitive e facili da utilizzare.Design Goal

1.2.1. Trade-off

- **Modificabilità vs Performance**
  Il sistema, data la scelta dell'architettura three-tier chiusa, è orientato alla manutenibilità e leggibilità comportando un costo sui criteri di performance.

- **Performance vs Memoria**

  Per garantire un certo livello di performance verranno introdotte, a discapito della memoria, ridondanze per evitare operazioni costose.

- **Affidabilità vs Tempo di risposta**

  Il sistema sarà implementato in modo da dare priorità all'affidabilità rispetto ai tempi di risposta, in modo da garantire una risposta corretta e consistente.

1.3. Definizioni, acronimi e abbreviazioni

- **Definizioni**

  GuardaTV: nome del sistema.

  Contenuto: oggetto di interesse dell'utente, può essere un film o una serie TV.

  Lista: insieme di contenuti sul quale un utente loggato può effettuare varie operazioni [creazione, aggiunta, rimozione]

  Utente non registrato: utente che non si è precedentemente registrato al sistema GuardaTV, può effettuare ricerche e visualizzare contenuti presenti nel sistema.

  Utente loggato: utente che si è registrato al sistema ed ha effettuato il login al sistema GuardaTV, ciascuno sarà caratterizzato da: E-Mail, Password, Nickname. Può creare liste, aggiungere contenuti a liste, recensire contenuti ed effettuare ricerche.

  Recensione: un utente registrato ha la possibilità di recensire un contenuto presente nel sistema. Una recensione è caratterizzata da un punteggio e una descrizione testuale.

  Utente amministratore: oltre ai permessi concessi ad un utente loggato, l'utente amministratore può aggiungere nuovi contenuti al sistema e rimuovere recensioni.

- **Acronimi**

  SDD: System Design Document

  DBMS: DataBase Management System

  CD: Class diagram

  REQ: Requisito

  SC: Scenario

1.4. Riferimenti

Object-Oriented Software Engineering Using UML, Patterns, and Java - 3rd Edition

1.5. Panoramica

Il SDD è diviso in quattro capitoli:

1. È composto dall'introduzione del sistema, i design goals e un elenco di definizioni e acronimi.
2. Contiene la descrizione della decomposizione in sottosistemi.
3. Contiene il controllo degli accessi.
4. Contiene le condizioni limite.

2. Architettura Software

2.1. Overview

Il sistema GuardaTV è un'applicazione web; utilizziamo un'architettura closed Three-tier, ovvero le funzionalità sono separate e suddivise in tre livelli, in comunicazione tra loro, in modo da separare la logica di presentazione dalla logica di business e ridurre l'accoppiamento tra i sottosistemi.

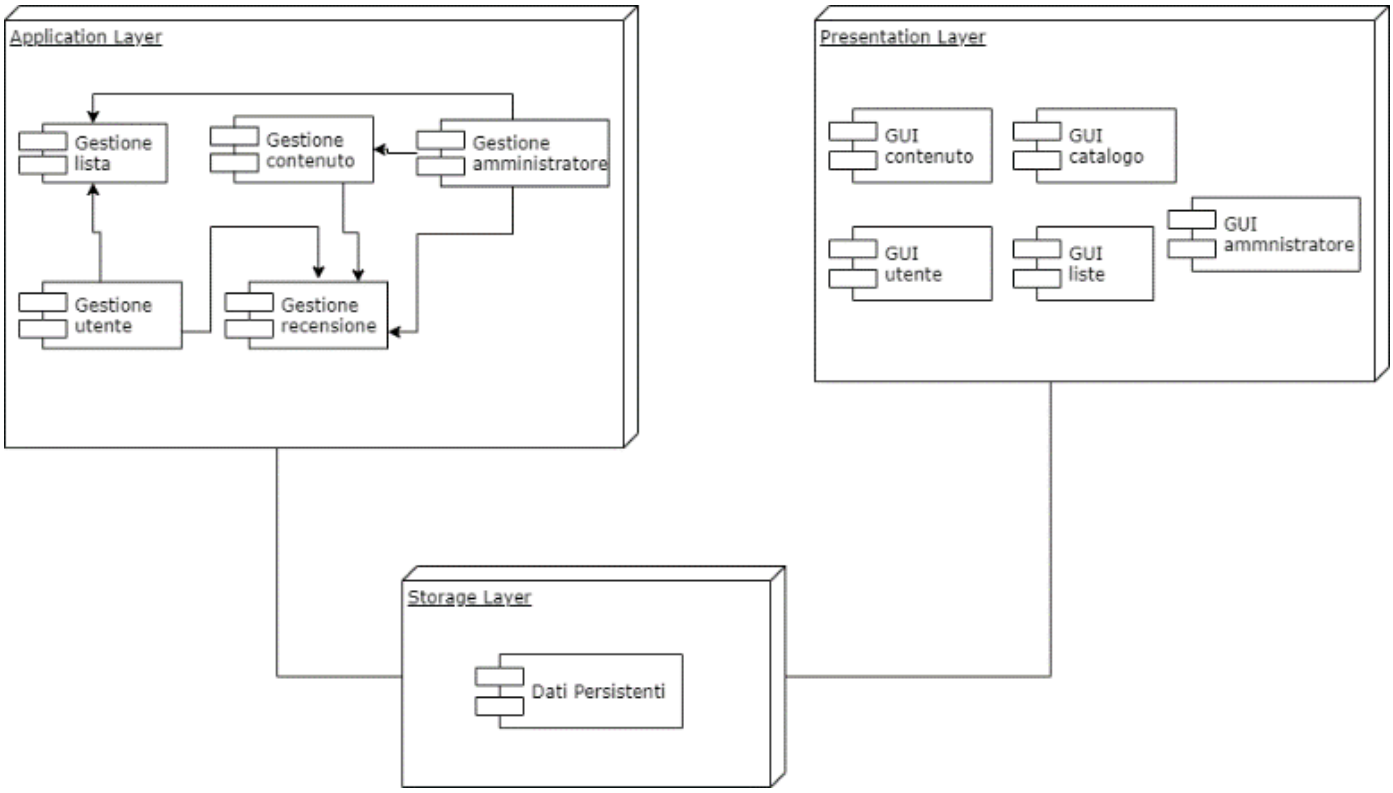
2.2. Subsystem decomposition

La decomposizione prevista è in tre layer che si occupano di gestire aspetti e funzionalità differenti:

1. Storage: Memorizzazione e gestione dei dati persistenti;

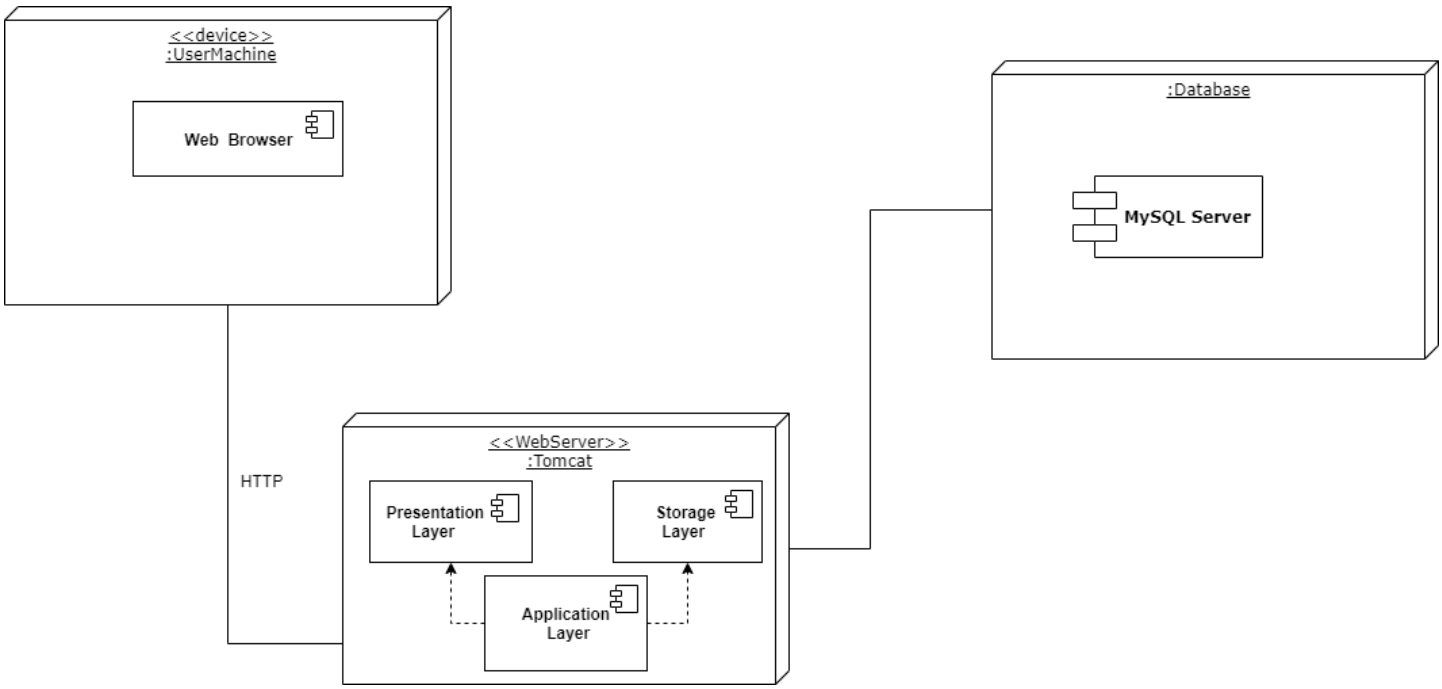
2. Application: Gestione dello scambio dei dati tra i sottosistemi ed implementa la logica applicativa;
3. Presentation: Raccoglie e gestisce elementi di interfaccia grafica e gli eventi generati su di essi;

Il presentation layer e l'application logic layer risiederanno sulla stessa istanza di application server; tuttavia, durante eventuali sviluppi futuri, sarà possibile separare facilmente questi due strati su due istanze distinte dello stesso application server.
Sono stati individuati cinque principali gruppi di funzionalità che il sistema deve supportare: funzionalità riguardanti gli account utenti, funzionalità riguardanti le recensioni, funzionalità riguardanti la gestione dei contenuti , funzionalità riguardanti le liste ed infine le funzionalità amministrative.Per ciascun gruppo di funzionalità,l'application logic layer deve implementare un servizi che lo soddisfi.



2.3. Hardware/Software mapping

WebServer: il server utilizzato è Apache Tomcat .
Interface layer: l'utente usufruisce del sistema GuardaTV tramite un'applicazione browser installata all'interno del suo calcolatore [e.g. Google Chrome, Mozilla Firefox, Opera, Microsoft Edge].



Application logic layer: le funzionalità del sistema sono state implementate mediante l'API Java Servlet.
Database Server: il DMBS usato è MySQL.

2.4. Persistent data managment

2.4.1. Identifiyng persistent object

Per la gestione dei dati persistenti del sistema e la loro memorizzazione si è optato per la scelta di un database relazionale. L'obbiettivo è quello di garantire brevi tempi di risposta ,garantire che i dati memorizzati siano consistenti e ridurre i limiti di spazio di archiviazione. E allo stesso tempo vogliamo un sistema che ci permetta di gestire in modo adeguato l'accesso concorrente ai dati, utilizzando un DBMS . Inoltre è possibile ripristinare lo stato del database in caso di danni software o hardware attraverso una copia dei dati fatta priodicamente.
Per garantire un occupazione dello spazio efficiente ed una maggior velocità, le immagini relative alle copertine dei contenuti digitali sono memorizzate nel file system del server, mentre nel database è memorizzato solo il riferimento all'immagine sotto forma di stringa.

GuardaTV tratta un insieme di oggetti che devono essere memorizzati.
L'oggetto UTENTE ,che memorizza i dati personali relativi all'utente , incluso un attributo booleano che identifica l'Utente Amministratore.

NOME	TIPO	CONSTRAINTS	KEY
Email	Varchar(256)	Not Null	Primary key
PasswordHash	Char(40)	Not Null	
Salt	Varchar(256)	Not Null	
DataDiNascita	Date	Not Null	
Username	Varchar(50)	Not Null	
Administrator	Boolean (default false)	Not Null	

L'oggetto CONTENUTO, che memorizza tutti i campi relativi al contenuto.

NOME	TIPO	CONSTRAINTS	KEY
Id	Varchar(33)	Not Null	Primary key
Titolo	Varchar(50)	Not Null	
Descrizione	Varchar(255)	Not Null	
Regista	Varchar(50)	Not Null	
Durata	Int(10)	Not Null	
DataDiUscita	Date	Not Null	

ImmagineDelContenuto	Varchar(255)		
VideoTrailer	Varchar(255)		
Film	Boolean (default true)	NotNull	
Stagioni	Int(default null)		
Puntate	Int(default null)		

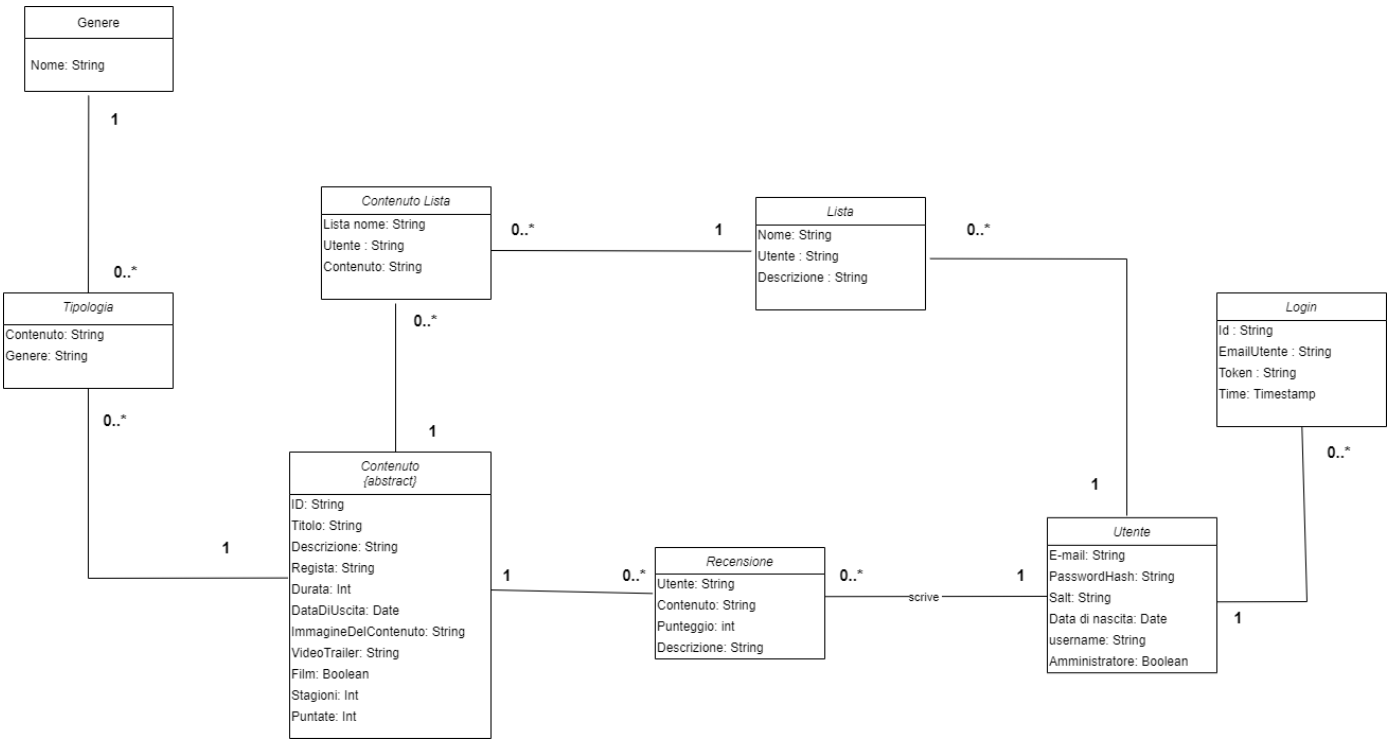
L'oggetto RECENSIONE, che memorizza tutte le recensioni fatte dagli utenti.

NOME	TIPO	CONSTRAINTS	KEY
Utente	Varchar(256)	Not Null	Primary key, Foreign key
Contenuto	Varchar(33)	Not Null	Primary key, Foreign key
Punteggio	Int	Not Null	
Descrizione	Varchar(256)		

E l'oggetto LISTA, che memorizza una lista di contenuti digitali selezionati e categorizzati piacere dall'utente.

NOME	TIPO	CONSTRAINTS	KEY
Nome	Varchar(50)	Not Null	Primary key
Utente	Varchar(256)	Not Null	Primary key, Foreign key
Descrizione	Varchar(255)	Not Null	

-Class Diagram



2.4.2. Storage strategy

La nostra più alta priorità sta nell'offrire all'utente un catalogo di contenuti digitali dotati di descrizioni e recensioni sempre aggiornate.

L'Utente , il Contenuto e la Recensione saranno memorizzati in un database, ad eccezione delle immagini relativi ai contenuti, che verranno salvate sul file system del server per garantire maggiore efficienza.

La Lista sarà memorizzata nel database per gli utenti loggati, mentre per gli utenti non autenticati verrà salvata nel file system del dispositivo in attesa di una successiva autenticazione per effettuare la sincronizzazione delle liste.

Per garantire la sicurezza dei dati sensibili degli utenti, le password verranno salvate solo dopo aver effettuato l'hashing del testo crittografato con aggiunta di salting.

Il database sviluppato sarà un database relazionale implementato utilizzando MySQL.

2.5. Access control and security

GuardaTV è un sistema multi-utente, ci sono diversi attori che hanno il permesso di eseguire diverse operazioni su vari insiemi di oggetti.

Le informazioni sensibili degli utenti vanno trattate in maniera sicura; in particolare, per la memorizzazione delle password degli utenti sulla base di dati, queste passeranno per la funzione di password hashing irreversibile bcrypt, il cui risultato sarà ultimamente persistito. Una funzione di hashing irreversibile prende come argomento una stringa da cui ne viene costruita una nuova di lunghezza fissa, restituita come risultato, attraverso la quale non è possibile risalire alla stringa originaria. Le funzioni di password hashing sono idempotenti, ovvero per lo stesso input producono sempre lo stesso risultato, il che ha degli effetti collaterali indesiderati: ad esempio, se due utenti utilizzano la stessa password, questa viene memorizzata sulla base di dati utilizzando la stessa sequenza prodotta dalla funzione di hashing; ciò presenta dei problemi per la sicurezza perché, nell'eventualità in cui un utente malintenzionato dovesse riuscire ad accedere ai record degli utenti memorizzati sulla base di dati, questo si accorgerebbe di più sequenze uguali nei campi delle password degli utenti, e potrebbe di conseguenza trarre la conclusione che le password utilizzate da questi siano sequenze notoriamente insicure (o, in altre parole, brutti esempi di password come password123 o simili) ed eventualmente ottenere l'accesso a più account. Per aggirare questo problema vengono introdotti i salt, ovvero ulteriori sequenze di caratteri, generati casualmente, che vengono combinate alla password dell'utente prima di passarla per la funzione di hashing; ciò ha l'effetto di garantire che, anche nel caso in cui più utenti dovessero utilizzare la stessa password, le sequenze memorizzate sulla base di dati siano teoricamente tutte distinte tra loro. Esiste la possibilità che la funzione di hashing generi la stessa sequenza per password diverse, o che dei salt generati casualmente possano essere identici per la stessa password, annullando quindi il vantaggio dell'introduzione dei salt per il gruppo di password affetto, ma la probabilità che ciò avvenga è talmente bassa da risultare trascurabile.

Per poter usufruire della maggior parte delle funzionalità offerte dal sistema, gli utenti dovranno autenticarsi. L'autenticazione avviene fornendo al sistema la combinazione di nome utente e password associate all'account a cui ci si vuole autenticare.

Per gestire meglio il controllo degli accessi abbiamo creato una tabella aggiuntiva nel Database intitolata "Login" ,strettamente collegata alla tabella "Utente". Questa contiene un identificativo ed un token ,l'email dell'utente che svolge l'accesso e il timestamp , che memorizza l'orario in cui è avvenuto l'accesso per ,eventualmente, gestire la scadenza della sessione. Per schematizzare al meglio il contollo degli accessi abbiamo suddiviso per tipologia di utente le azioni consentite, al fine di ottenere una visione compatta e dettagliata grazie ad una matrice degli accessi riportata di seguito:

	Contenuto	**Lista**	**Recensione**	**Utente**

Utente non registrato	Visualizzazione		Visualizzazione	Registrazione
Utente loggato	Visualizzazione	Creazione Modifica Rimozione	Creazione	Visualizzazione Modifica
Utente amministratore	Creazione Modifica Rimozione Visualizzazione		Rimozione	

2.6. Global software control

I client effettuano richieste HTTP all'application server, che tramite un thread dedicato, in modo da garantire un'interazione concorrente con tutti gli utenti connessi, li re-indirizza sugli appositi endpoint di dispatching che si occupano di fornire le view; questi le elaborano e rispondono ai client con le view richieste, generate dinamicamente; tramite appositi elementi delle view, i client effettuano ulteriori richieste HTTP asincrone all'application server, che nuovamente le smista agli appositi endpoint di controllo che le elaborano e rispondono di conseguenza; nell'elaborazione delle richieste, l'application server interagisce continuamente con la base di dati sottostante, interrogandola o aggiornandone il contenuto.

2.7. Boundary Condition

2.7.1. Avvio del sistema

Lo start-up del sistema prevede l'avvio del web server nel quale il sistema è installato e l'avvio del DBMS per accedere ai dati persistenti memorizzati nel database. Quando sia il web server che il DBMS sono in esecuzione, il sistema carica in memoria centrale le servlet principali attraverso le quali gli utenti possono effettuare le operazioni. Dopo l'avvio del sistema gli utenti possono interagire con esso.

2.7.2. Shut Down

Quando il sistema deve essere arrestato, il gestore del sistema termina l'esecutivo del web server. Quando ciò avviene tutte le risorse che il sistema utilizza (connessione al database e connessione alla rete) vengono rilasciate e nessun utente potrà più connettersi al sistema.

2.7.3. Fallimento

1. Nel caso in cui si presentasse un'interruzione inaspettata dell'alimentazione, non vi sono metodi per ripristinare lo stato del sistema precedente allo spegnimento non voluto. Qualsiasi transazione con il database viene annullata e viene ripristinato lo stato consistente più recente delle informazioni persistenti
2. In caso di guasti dovuti al sovraccarico di informazioni al database, la rete viene congestionata. Il Web Server in questo stato inviterà tutti i clienti connessi a riprovare le operazioni effettuate in un secondo momento.
3. Nel caso di una chiusura inaspettata del software, dovuta ad errori avvenuti durante la fase di implementazione, il server risponderà con una pagina di errore
4. Nel caso di ricezione di informazioni errate da parte di un utente, o che non permettono la corretta esecuzione di un operazione, il server risponderà con un messaggio di errore
5. Nel caso di un errore critico dell'hardware non è prevista una soluzione.

2.7.4. Use Cases

Identificativo UC_SU_	Start-up	Data	29/11/2021
		Vers.	0.00.001
		Autore	Silvio Pastore

Descrizione		Funzionalità per l'avvio del sistema
Attore Principale		Amministratore
Entry Condition		L'amministratore visualizza la console per effettuare l'avvio del sistema
Exit condition On success		Il sistema è avviato e funzionante
Exit condition On failure		Il sistema non è avviato (?)
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	**Amministratore:**	invia il comando di avvio
2	**Sistema:**	esegue le opportune procedure di avvio, attiva i server ed i servizi in remoto rendendoli disponibili alle richieste esterne
Note		
2	**PUNTO DA DISCUTERE:** Cosa fare quando il sistema fallisce la procedura di avvio?	

Identificativo UC_SD_	Shutdown	Data	29/11/2021
		Vers.	0.00.001
		Autore	Silvio Pastore

Descrizione		Funzionalità per l'arresto del sistema
Attore Principale		Amministratore
Entry Condition		L'amministratore visualizza la console per effettuare la terminazione del sistema
Exit condition On success		Il sistema è arrestato correttamente
Exit condition On failure		Errore durante l'arresto del sistema
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	**Amministratore:**	invia il comando di arresto
2	**Sistema:**	esegue le opportune procedure di arresto, controlla eventuali richieste in sospeso, salva i dati necessari, disattiva i servizi e il server.
I Scenario/Flusso di eventi di ERRORE: il sistema non riesce ad effettuare il salvataggio dei dati		
2.1	**Sistema:**	effettua un salvataggio di emergenza sul file system che verrà poi controllato alla prossima accensione (priorità bassa)
Note		
2.1	**PUNTO DA DISCUTERE:** Cosa fare quando il sistema fallisce il salvataggio dei dati?	
2.1	**PUNTO DA DISCUTERE:** Cosa fare quando il sistema fallisce l'arresto del sistema?	

3. Subsystem Services

Utente non registrato:

- Gestione utente: l'utente non registrato può effettuare la registrazione al sistema inserendo le proprie credenziali
- Gestione contenuto: l'utente non registrato può ricercare e visualizzare le informazioni relative ai contenuti presenti

Utente registrato:

- Gestione utente: l'utente registrato può effettuare l'accesso al sistema utilizzando le proprie credenziali. Può anche visualizzare le proprie informazioni personali
- Gestione contenuto: l'utente registrato può ricercare e visualizzare le informazioni relative ai contenuti presenti
- Gestione recensione: l'utente registrato può recensire un contenuto presente nel sistema
- Gestione lista: l'utente registrato può creare una lista di contenuti personalizzata. Può aggiungere e rimuovere contenuti dalla lista.

Utente admin:

- Gestione account: l'utente admin può effettuare l'accesso al sistema utilizzando le proprie credenziali. Può anche visualizzare le proprie informazioni personali
- Gestione contenuto: l'utente può ricercare e visualizzare le informazioni relative ai contenuti presenti.
- Gestione amministratore: l'utente admin può aggiungere un nuovo contenuto al sistema e rimuovere una recensione di un utente registrato
- Gestione recensione: l'utente può recensire un contenuto presente nel sistema.
- Gestione lista: l'utente può creare una lista di contenuti personalizzata. Può aggiungere e rimuovere contenuti dalla lista.