

SDD

Progetto IS

NetSection

Riferimento	
Versione	2.0
Data	09/12/2022
Destinatario	Docenti di Ingegneria del software 2022/23
Presentato da	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
Approvato da	

Revision History

Data	Versione	Descrizione	Autori
30/11/2022	1.0	Creazione SDD e aggiunta Design Goals e Trade-offs.	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
05/12/2022	1.1	Aggiunta hardware/software mapping, persistent data management, controllo accesso e sicurezza, controllo software globale e boundary conditions.	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
07/12/2022	1.2	Aggiunta e revisione dei sottosistemi.	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
09/12/2022	2.0	Aggiornamento persistent data management, aggiunta Class Diagram ristrutturato e aggiunta Design Goals	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe

Sommario

1. [Introduzione](#)
 - 1.1. [Obiettivo del sistema](#)
 - 1.2. [Design Goals e Trade-offs](#)
 - 1.2.1. Prestazione vs affidabilità
 - 1.2.2. Prestazione vs Costi
2. [Architettura software simili](#)
3. [Architettura software proposta](#)
 - 3.1. [Decomposizione sottosistema](#)
 - 3.2. [Hardware/software mapping](#)
 - 3.3. [Persistent data management](#)
 - 3.4. [Controllo accesso e sicurezza](#)
 - 3.5. [Controllo software globale](#)
 - 3.6. [Boundary conditions](#)

1. Introduzione

Nel presente documento verranno definiti quelli che sono gli obiettivi in termini di design, prestazioni ed affidabilità, verrà fornita una descrizione del database, come verranno gestiti i fallimenti e come sono definiti i diversi livelli di permessi alla piattaforma.

1.1. Obiettivo del sistema

Netflix ha come obiettivo principale è quello di consolidare le fan-base attraverso uno strumento che consente agli utenti di esprimere le loro opinioni su film e serie disponibili nel catalogo.

1.2. Design Goals e Trade-offs

Nella seguente tabella vengono elencati i design goals della piattaforma da realizzare.

RANK	ID	DESCRIZIONE	CATEGORIA	RNF di origine
	DG_1	Il sistema deve garantire la divisione degli utenti per impedire agli utenti di compiere azioni che esulano la loro categoria.	Affidabilità	RNF_RDA_1
	DG_2	Non appena si verifica un errore, il sistema procede, nel giro di massimo un minuto, con la segnalazione del problema all'utenza e con la manutenzione del sito.	Affidabilità	RNF_RDA_2
	DG_3	Il sistema deve garantire il corretto funzionamento del sito sulla maggior parte dei browser tramite l'uso di codice html che non crei conflitti sui diversi browser.	End user criteria	RNF_RDA_3
	DG_4	Il sistema deve garantire un tempo di risposta tra i 5 e i 10 secondi.	Performance	RNF_RDP_1
	DG_5	Il sistema deve gestire gli eventuali condizioni di errore generati da input invalidi.	End user criteria	RNF_RDIP_1

	DG_6	Il sistema deve avere a disposizione uno spazio di circa 1 Terabyte per poter conservare tutti i dati.	Performance	RNF_RDIP_2
--	------	--	-------------	------------

1.2.1. Prestazione vs Affidabilità

Dato che si vuole dare all'utente un'esperienza di discussione quanto più vicina alla realtà, vengono preferite le prestazioni piuttosto che un maggiore controllo dei dati.

1.2.2 Prestazione vs Costi

Considerando il budget stanziato, si preferisce dare più importanza allo sviluppo del sistema e alla revisione.

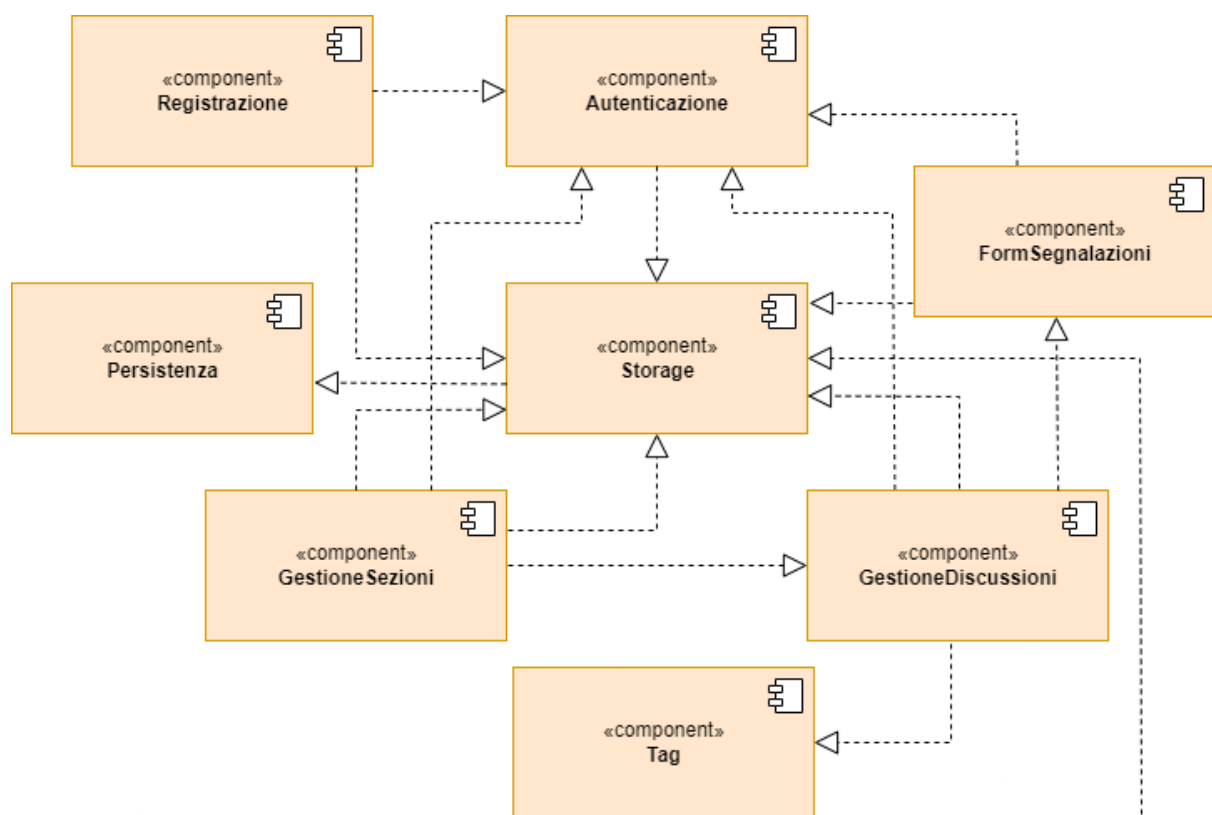
2. Architetture software simili

La piattaforma si ispira a forum esistenti come Reddit o più specifico per i commenti dei film al forum Filmup. Allo stesso tempo l'interfaccia utente riprende quella della piattaforma di Netflix in quanto in collegamento fra di loro. L'organizzazione delle sezioni nelle pagine è simile a quella di Netflix mentre da reddit si prende spunto per l'organizzazione delle discussioni e dei commenti degli utenti.

3. Architettura software proposta

In questa sezione viene definita l'architettura, che verrà usata per la realizzazione della piattaforma, e le sue componenti interne.

3.1. Decomposizione sottosistema



L'intero sistema è stato diviso in componenti, ognuna delle quali descrive un singolo oggetto del sistema, le componenti individuate rappresentano i sottosistemi che sono:

Registrazione: si occupa della registrazione utente (normale utente o utente Netflix).

Autenticazione: permette all'utente di fare il login, logout o modificare i propri dati.

Storage: prende i dati degli altri sottosistemi per trasmetterli al sottosistema di persistenza.

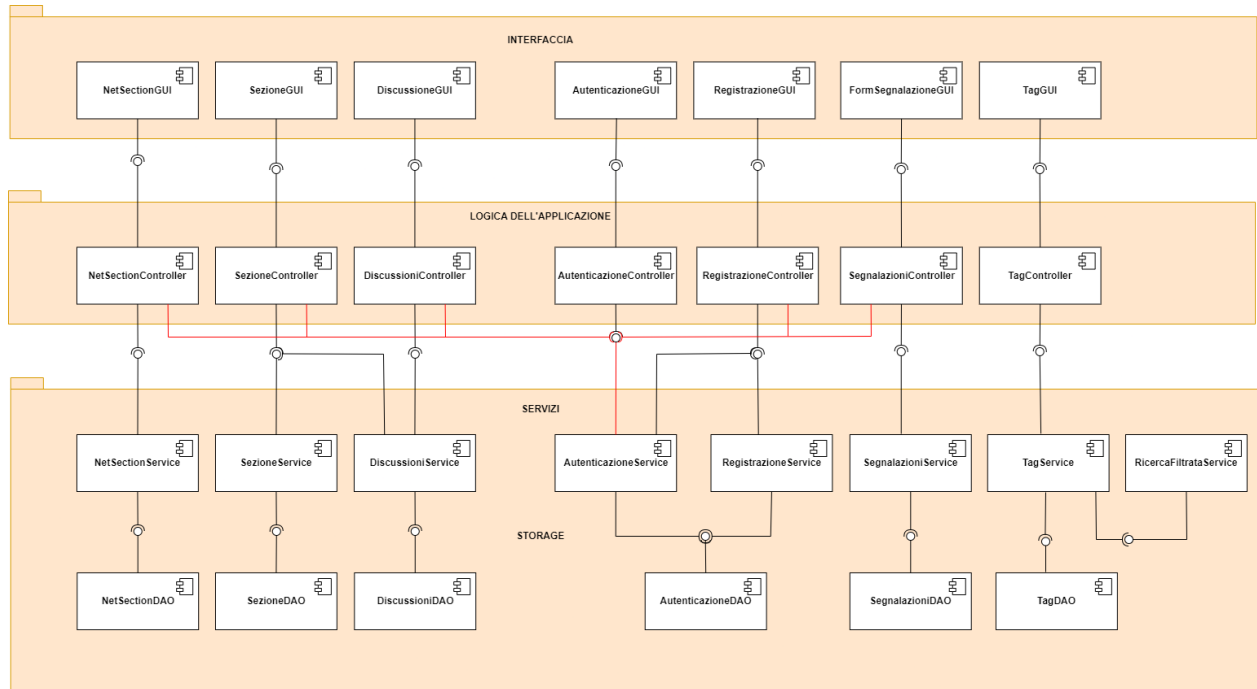
Persistenza: conserva i dati degli altri sottosistemi.

FormSegnalazioni: gestisce le segnalazioni dei commenti nelle discussioni, generate da utenti registrati.

GestioneSezioni: permette di gestire ciò che riguarda le sezioni. La gestione può esser fatta solo da un utente autenticato come "admin".

GestioneDiscussioni: permette di gestire ciò che riguarda le discussioni. La gestione può esser fatta solo da un utente autenticato come “moderatore”.

Tag: permette di effettuare azioni importanti per l’utente, come ad esempio la ricerca filtrata.



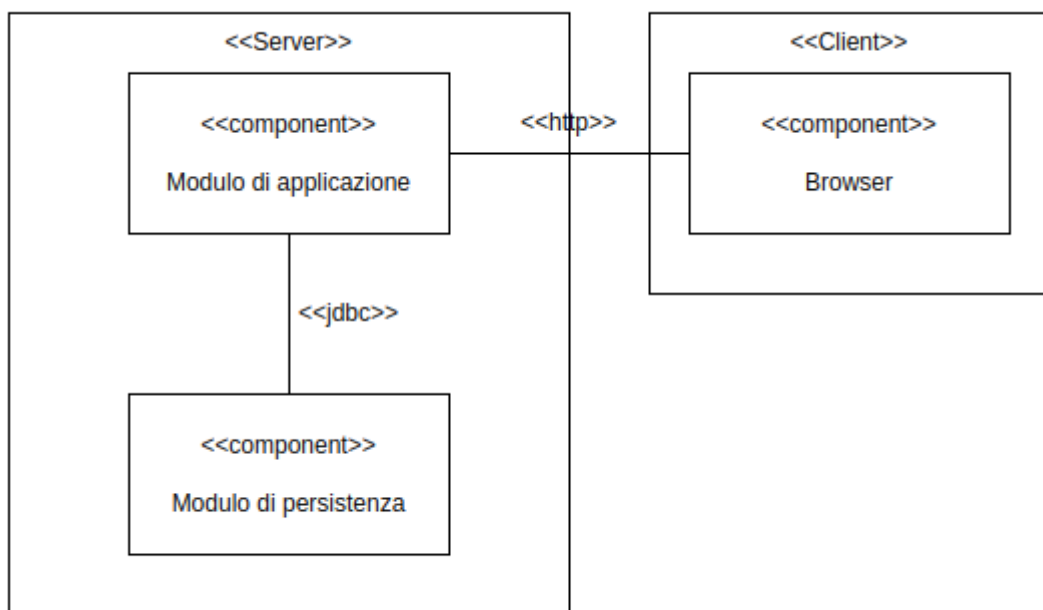
Il sistema è stato decomposto in tre sottosistemi ovvero storage, logica dell'applicazione e interfaccia, pertanto la scelta dell'architettura ricade in una three-tier, questa scelta dipende dal fatto che si vuole mantenere la gestione dei dati persistenti, la logica e la parte di presentazione indipendenti fra di loro.

In particolare l'interfaccia comprende l'Admin GUI che rappresenta l'interfaccia resa disponibile all'admin che comprende la pagina di amministrazione dell'admin, il Visitatore GUI che rappresenta l'interfaccia visibile all'utente non registrato alla piattaforma che comprende l'home page, le discussioni e la pagina delle sezioni piattaforma e l'Utente Registrato GUI che rappresenta l'interfaccia resa disponibile all'utente registrato con la sua pagina di amministrazione, la pagina di scrittura delle recensioni e la pagina per le segnalazioni.

La parte di Logica dell'applicazione comprende invece le componenti di gestione degli elementi principali della piattaforma come le sezioni, discussioni, utenti, segnalazione e tag. Infine la parte di storage che è costituita dal database sottostante alla piattaforma che gestisce la definizione e manipolazione dei dati.

3.2. Hardware/Software mapping

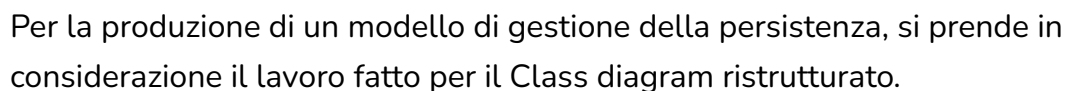
Il sistema si compone di tre livelli ovvero il livello che si occupa della logica e gli elementi di gestione del sistema, il livello di gestione dei dati persistenti che si interfaccia con il database ed infine il livello di presentazione che si occupa di organizzare gli oggetti nelle pagine di visualizzazione del client.



La parte hardware è composta da un unico server che contiene sia il modulo di applicazione che il modulo di persistenza. Il server deve supportare:

- Apache Tomcat per l'esecuzione dell'applicativo web che si intende sviluppare;
- MySQL per la definizione e manipolazione dei dati persistenti;

(l'immagine è un link)



- Non vengono considerate per il mantenimento di dati persistenti, quindi non figurano esplicitamente all'interno del modello relazionale.

- 1.1.1. Kick, Eliminare(tramite Admin e UtenteRegistrato), Eliminare(tramite Admin e Sezione), Eliminare(tramite Utente e Discussione), Disiscrizione(tramite Utente e Discussione), vengono rappresentati da rimozione di istanze da altre relazioni. Ad esempio

Disiscrizione è rappresentata con la perdita di una specifica istanza Iscrizione, che fa riferimento ad uno specifico UtenteRegistrato e una Discussione.

- 1.1.2. Creare(tramite Admin e Sezione) invece viene rappresentata con l'aggiunta di una nuova istanza di Sezione. Per la definizione che abbiamo dato alla Sezione non è necessario mantenere dati sull'Admin che la crea.

2. Definizione degli utenti:

- 2.1. L'utente registrato può essere un utente netflix, il quale è dotato di una emailNetflix, che fa riferimento alla mail di un profilo Netflix, e una relativa passwordNetflix. Un utente netflix è istanziato ad ogni operazione di Registrazione utente Netflix sulla base dell'UtenteRegistrato che ha svolto l'operazione. L'UtenteNetflix è che un particolare UtenteRegistrato che ha particolari privilegi.
 - 2.2. Siccome un qualunque utente registrato può fungere il ruolo di moderatore, abbiamo deciso di accorpare il moderatore nell'UtenteRegistrato. Un moderatore è quindi definito come un qualunque utente registrato partecipante ad almeno una associazione moderare.
3. Per la segnalazione dei problemi definiamo un problema con l'utente che lo ha rilevato e la data della sua rilevazione.
 4. Un genere è identificato solo dal nome e sono preesistenti all'interno del DB.
 5. Per l'operazione di scelta generi, siccome un utente può favorire più genere, e siccome i generi possono essere favoriti da più utenti, usiamo istanze di una relazione Preferire identificata sia dall'utente che dal genere.
 6. l'operazione di visualizzazione delle discussioni rapida è resa possibile con l'uso della relazione Iscrizione. Essa mantiene i dati riguardo all'iscrizione di un utente registrato ad una discussione.
 7. L'elezione di un moderatore viene gestita tramite la creazione di una relazione Moderare.

Modello Relazionale:

UtenteRegistrato(id, username, email, password, età, immagine)

UtenteNetflix(idUtente, emailNetflix, passwordNetflix)

Problema(idUtente, dataSottomissione, natura, contenuto)

Genere(nome)

Preferire(idUtente, genere)

Sezione(idSezione, immagine, titolo, descrizione)

Appartenere(idSezione, genere)
 Discussione(sezione, titolo, creatore, tags, immagine, dataCreazione)
 Iscrizione(idUtente, sezione, discussione)
 Moderare(idUtente, sezione, discussione)
 Commento(dataScrittura, creatore, sezione, discussione, contenuto)
 Segnalazione(dataSegnalazione, creatoreSegnalazione, dataCommento,
creatoreCommento, sezione, discussione, natura, contenuto)
 Admin(idAdmin, username, email, password)

Vincoli di Identità Referenziale:

UtenteNetflix(idUtente) ha VIR con UtenteRegistrato(id);
 Problema(idUtente) ha VIR con UtenteRegistrato(id);
 Preferire(idUtente) ha VIR con UtenteRegistrato(id);
 Preferire(genere) ha VIR con Genere(nome);
 Appartenere(idSezione) ha VIR con Sezione(idSezione);
 Appartenere(genere) ha VIR con Genere(nome);
 Discussione(sezione) ha VIR con Sezione(idSezione);
 Discussione (creatore) ha VIR con UtenteNetflix(idUtente);
 Iscrizione(idUtente) ha VIR con UtenteRegistrato(id);
 Iscrizione(discussione) ha VIR con Discussione(titolo);
 Iscrizione(sezione) ha VIR con Discussione(sezione);
 Moderare(idUtente) ha VIR con UtenteRegistrato(id);
 Moderare(discussione) ha VIR con Discussione(titolo);
 Moderare(sezione) ha VIR con Discussione(sezione);
 Commento(creatore) ha VIR con UtenteRegistrato(id);
 Commento(discussione) ha VIR con Discussione(titolo);
 Commento(sezione) ha VIR con Discussione(sezione);
 Segnalazione(creatoreSegnalazione) ha VIR con UtenteRegistrato(id);
 Segnalazione(dataCommento) ha VIR con Commento(data);
 Segnalazione(creatoreCommento) ha VIR con Commento(creatore);
 Segnalazione(discussione) ha VIR con Commento(discussione);
 Segnalazione(sezione) ha VIR con Commento(sezione);

3.4. Controllo accesso e sicurezza

Il controllo degli accessi viene eseguito mediante il sistema di autenticazione della piattaforma stessa che attraverso dei controlli sui dati di profilo inseriti assegna un gruppo di permessi all'utente appena loggato per poter eseguire solo le operazioni a lui permesse.

Nella tabella sottostante vengono definiti i gruppi di permessi per ogni tipo di utente e le relative operazioni.

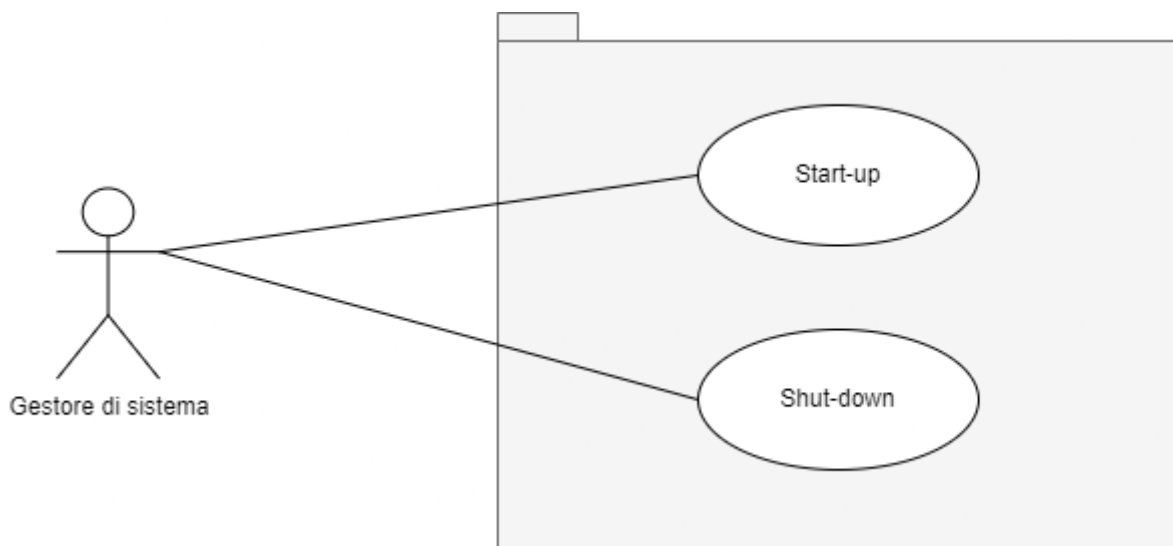
Funzionalità/ Utente	Funzionalità di base	Funzionalità discussioni	
Visitatore	-visualizzazione discussioni e sezioni; -registrazione; -registrazione tramite netflix;	-ricerca filtrata discussione;	
Utente Registrato	-visualizzazione discussioni e sezioni; -ricerca filtrata discussione; -modifica dati personali; -scelta generi preferiti; -accesso rapido alle discussioni;	-iscriversi ad una discussione; -scrittura commenti; -votare un commento; -disiscriversi da una discussione; -ricerca filtrata discussione;	-inviare una segnalazione ad utente e commento; -segnalare un utente;
Utente Netflix	-visualizzazione discussioni e sezioni; -ricerca filtrata discussione; -elezione di un nuovo moderatore;-modifica dati personali; -scelta generi preferiti; -accesso rapido alle discussioni;	-iscriversi ad una discussione; -scrittura commenti; -votare un commento; -disiscriversi da una discussione; -ricerca filtrata discussione; -creazione di una discussione; -elezione di un nuovo moderatore;	-inviare una segnalazione ad utente e commento; -segnalare un utente;
Moderatore	-visualizzazione discussioni e sezioni;	-iscriversi ad una discussione;	-inviare una segnalazione ad utente

	<ul style="list-style-type: none"> -ricerca filtrata discussione; -modifica dati personali; -scelta generi preferiti; -accesso rapido alle discussioni; 	<ul style="list-style-type: none"> -scrittura commenti; -votare un commento; -disiscriversi da una discussione; -ricerca filtrata discussione; -elezione di un nuovo moderatore; 	<ul style="list-style-type: none"> e commento; -segnalare un utente; -cacciare un utente da una discussione; -cancellare un commento;
Amministratore			<ul style="list-style-type: none"> -aggiornamento del catalogo; -ban di un utente dalla piattaforma;

3.5. Controllo software globale

Il sistema adotta un controllo di flusso globale di tipo event-driven, con event-driven si intende l'esecuzione di ogni operazione al verificarsi di un evento specifico. Il vantaggio dell'uso di un approccio di tipo event-driven è che gli eventi sono indipendenti dalla tecnologia utilizzata e l'hardware sottostante al sistema.

3.6. Boundary conditions



Start-up:

Identificativo <i>UC_BC_01</i>	Start-up	Data	05/12/2022
		Versione	1.0
		Autore	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
Descrizione	Lo Use Case descrive le funzionalità all'avvio del sistema.		
Attore Principale	Gestore di sistema		
Attori Secondari	NA		
Entry Condition	Il gestore di sistema richiede alla macchina l'avvio del sistema.		
Exit Condition <i>On Success</i>	Il sistema viene avviato correttamente.		
Exit Condition <i>On Failure</i>	Il sistema non viene avviato.		
Rilevanza/User Priority	Alta		
Frequenza Stimata	12/anno o 1/mese		
FLUSSO EVENTI PRINCIPALE/MAIN SCENARIO			
1	Gestore di sistema	Avvia MySQL.	
2	Gestore di sistema	Avvia Tomcat.	
3	Sistema	Il sistema si avvia correttamente e dà il benvenuto all'utente.	

Shut-down:

Identificativo <i>UC_BC_02</i>	Shut-down	Data	05/12/2022
		Versione	1.0
		Autore	Civetta Christian, De Luca Salvatore Michele, Grano Giuseppe
Descrizione	Lo Use Case descrive le funzionalità di terminazione del sistema.		
Attore Principale	Gestore di sistema		
Attori Secondari	NA		
Entry Condition	Il gestore di sistema richiede alla macchina la terminazione del sistema.		
Exit Condition <i>On Success</i>	Il sistema viene terminato correttamente.		
Exit Condition <i>On Failure</i>	Il sistema non viene terminato.		
Rilevanza/User Priority	Alta		
Frequenza Stimata	12/anno o 1/mese		
FLUSSO EVENTI PRINCIPALE/MAIN SCENARIO			
1	Gestore di sistema	Termina MySQL.	
2	Gestore di sistema	Termina Tomcat.	
3	Sistema	Il sistema termina correttamente.	

Failure:

Il fallimento può avvenire sia a livello hardware che a livello software:

- **Hardware:** la perdita parziale o totale dei dati causata da un problema di hardware non è tutelata da un backup di sicurezza a causa del budget non sufficiente.
- **Software:**
 - impossibile connettersi al database: viene visualizzato un messaggio di errore con l'invito a riprovare più tardi.
 - impossibile registrarsi con l'account Netflix: viene visualizzato un messaggio di errore con l'invito a contattare l'assistenza di Netflix o a riprovare più tardi.
 - eventuali problemi legati a operazioni in corso bloccate inaspettatamente verranno terminate.
 - i dati permanenti, come la creazione di una discussione già avvenuta, vengono salvati nel Database, dunque non c'è possibilità di perderli.