

System Design Document

Revision History

Data	Versione	Cambiamenti	Autori
20/12/2019	0.1	Aggiunta sezione 1, 2 e 3.1	Francesco Lauro
22/12/2019	0.2	Aggiunta sezione 3.2, 3.3,3.5, 3.6	Antonio Lodato Federico
23/12/2019	0.3	Aggiunta sezione 3.4, 3.7, e 4	Francesco Lauro
03/01/2020	0.4	Revisione intero documento	Antonio Lodato Federico Francesco Lauro

1. Introduction
 - 1.1. Purpose of the system
 - 1.2. Design goals
 - 1.3. Definition, acronyms and abbreviations
 - 1.4. References
 - 1.5. Overview
2. Current software architecture
 - 2.1. Analisi delle piattaforme esistenti
3. Proposed system architecture
 - 3.1. Overview
 - 3.2. Subsystem decomposition
 - 3.2.1. Decomposizione in layer
 - 3.2.2. Decomposizione in sottosistemi
 - 3.3. Hardware/software mapping
 - 3.4. Persistent data management
 - 3.4.1. Selecting a storage management strategy
 - 3.4.2. Identifying persistent objects
 - 3.5. Access control and security
 - 3.6. Global software control
 - 3.7. Boundary condition
4. Subsystem services

1. Introduction

1.1 Purpose of the system

il sistema che si vuole realizzare ha come scopo quello di facilitare la ricerca delle informazioni relative al settore informatico. L'obiettivo è quello di realizzare un sistema che da un lato facilita la ricerca di informazioni agli utenti, dall'altro focalizza il proprio core sulla correttezza delle informazioni apprese dall'utente mediante una moderazione di quelle che sono le informazioni pubblicate sulla piattaforma. Inoltre, il sistema focalizza la propria attenzione anche sulla comprensione degli utenti delle informazioni apprese supportandoli appositi sistemi di comunicazione con gli autori. In questo modo offriamo quindi una piattaforma comune sia ad autori che utenti.

1.2 Design goals

Criteri di performance:

- Tempi di risposta: il sistema deve rispondere alle operazioni dell'utente in massimo 3 secondi anche se il sistema è sotto stress (picco di carico elevato). Il sistema deve essere capace di visualizzare le pagine in massimo 2 secondi nonostante le condizioni di connessione dell'utente per garantire una miglior esperienza di navigazione
- Memoria: la memoria dipende dalla quantità di richieste che il sistema riceve e dalla memoria utilizzata per mantenere le informazioni memorizzate nel DB (ovviamente quelle necessarie al funzionamento). Si cercherà di ottimizzare lo spazio occupato in memoria con ottimizzazione sul DB

Criteri di Affidabilità:

- Robustezza: il sistema informerà l'utente, attraverso l'utilizzo di appositi messaggi d'errore, in presenza di input che non rispettano i formati definite nelle tabelle presenti all'interno del RAD
- Affidabilità: il sistema deve garantire l'affidabilità dei servizi offerti; la piattaforma sarà sviluppata in modo da controllare tutti gli input inseriti sia da utenti che autori, e i processi di registrazione e autenticazione saranno gestiti in modo affidabile.
- Disponibilità: il sistema sarà disponibile all'utente secondo le sue esigenze ad eccezione dei periodi di manutenzione che verranno comunicati.
- Sicurezza: il sistema fornisce le principali tecniche per poter resistere ad attacchi come ad esempio l'SQLinjection

Criteri di costo:

- Costo di sviluppo: viene stimato un costo complessivo di 200 ore per lo sviluppo del sistema divise tra i vari componenti del team.
- Costo di distribuzione: viene stimato un costo di pochi minuti per le formazioni degli utenti per poter utilizzare la nuova piattaforma
- Costo di manutenzione: viene richiesto un costo di 24 ore per la correzione di problemi sia software che hardware

Criteri di manutenzione:

- Estensibilità: è possibile aggiungere nuove funzionalità in base alle esigenze dell'utente o delle nuove tecnologie
- Portabilità: l'utilizzo del sistema avviene mediante l'utilizzo di un browser quindi il sistema può funzionare su qualsiasi dispositivo avente un browser web. Inoltre, l'utilizzo di funzioni di base supportate all'interno di qualsiasi browser web facilita ancor di più la portabilità

Criteri di Usabilità:

- Utilità: il sistema migliora la ricerca e l'apprendimento delle informazioni basandosi soprattutto sulla loro correttezza supportando così gli utenti.
- Usabilità: l'utilizzo del sistema risulta molto semplice visto l'utilizzo di un'interfaccia user-friendly e mediante apposite sezioni di supporto presenti nell'interfaccia, da suggerimenti e campi d'esempio.

1.3 Definition,acronyms and abbreviations

DB: Database

RAD: Requirements Analysis document

DBMS: Database Management system

JDBC: Java Database Connectivity

SDD: system design document

1.4 Riferimenti

Il materiale di riferimento utilizzato per la realizzazione del progetto e per la stesura di questo stesso documento comprende:

Libro di Testo: Object-Oriented Software Engineering Bruegge, A.H. Dutoit.

Slide fornite dal Professore Andrea De Lucia reperibili sulla piattaforma e-learning

1.5 Overview

Il seguente documento SDD comprende le seguenti sezioni:

1. Introduzione: in questa sezione sarà presentata una breve descrizione degli obiettivi del sistema e vengono forniti i design goal. Successivamente vengono presentate le definizioni, acronimi e abbreviazioni per facilitare al lettore la comprensione di alcuni termini.
2. Architettura del sistema corrente: il suo scopo è fornire la descrizione dell'architettura del sistema, se esiste, da sostituire.
3. Architettura del sistema proposto: lo scopo di questa sezione è quello di documentare il modello di progettazione del nuovo sistema. E' diviso in sette sezioni: overview, subsystem decomposition, Hardware/software mapping , persistent data management, access control and security, global software control, boundary condition.
4. Servizi di sottosistemi: questa sezione descrive i servizi forniti da ciascun sottosistema.

2.Current software Architecture

Attualmente non esiste una piattaforma dove gli utenti possano reperire informazioni che sono verificate, agevolando nella ricerca e nella comprensione di esse. Infatti, la ricerca di informazioni prevede i seguenti passi: l'utente effettua una ricerca all'interno di un browser web dove vengono prodotti come risultato di ricerca informazioni che in alcuni casi possono essere errati. Inoltre nella maggior parte dei casi le informazioni ricercate sono difficili da individuare poiché molti articoli contenenti le informazioni desiderate sono scritti da autori sconosciuti. L'utente solitamente si sofferma sulla lettura di blog famosi e in molti casi sono costretti a ricerca estenuanti che non producono nemmeno i risultati desiderati e nel caso li producessero non viene assicurata nessuna garanzia che le informazioni siano corrette. Inoltre, gli utenti sono costretti in alcuni casi ad utilizzare software esterni per poter comunicare con gli autori delle informazioni. Quelli descritti sono varie situazioni in cui si può trovare un utente. Questo può essere agevolato fornendo una piattaforma comune, sia ad utenti che autori, dove poter reperire le informazioni in maniera semplice e fornirne meccanismi estremamente semplici per comunicare.

3. Proposed software architecture

3.1 Overview

Il sistema proposto è un application-web che segue il pattern MVC separando la parte di logica da quella dei dati e dalla logica di business. Si tratta di un'architettura multi-tier dove le varie funzionalità sono separate su più layer. Il sistema supporta la ricerca di informazioni focalizzandosi sulla loro correttezza garantendola grazie ad una fase di moderazione. Il sistema offre anche un semplice sistema di comunicazione. Il sistema sarà utilizzato da tre tipi di utenti: utenti, autori e moderatori. Tutti gli utenti potranno autenticarsi e registrarsi alla piattaforma. Ogni utente avrà ovviamente le proprie funzionalità in particolare gli utenti potranno visualizzare le informazioni presenti negli articoli, comunicare con gli autori e visualizzare gli eventi in programma. Un autore potrà organizzare eventi ma come principale funzione potrà richiedere la pubblicazione di un articolo che successivamente verrà sottoposto a moderazione da parte di un apposito moderatore esperto del settore.

3.2 Subsystem decomposition

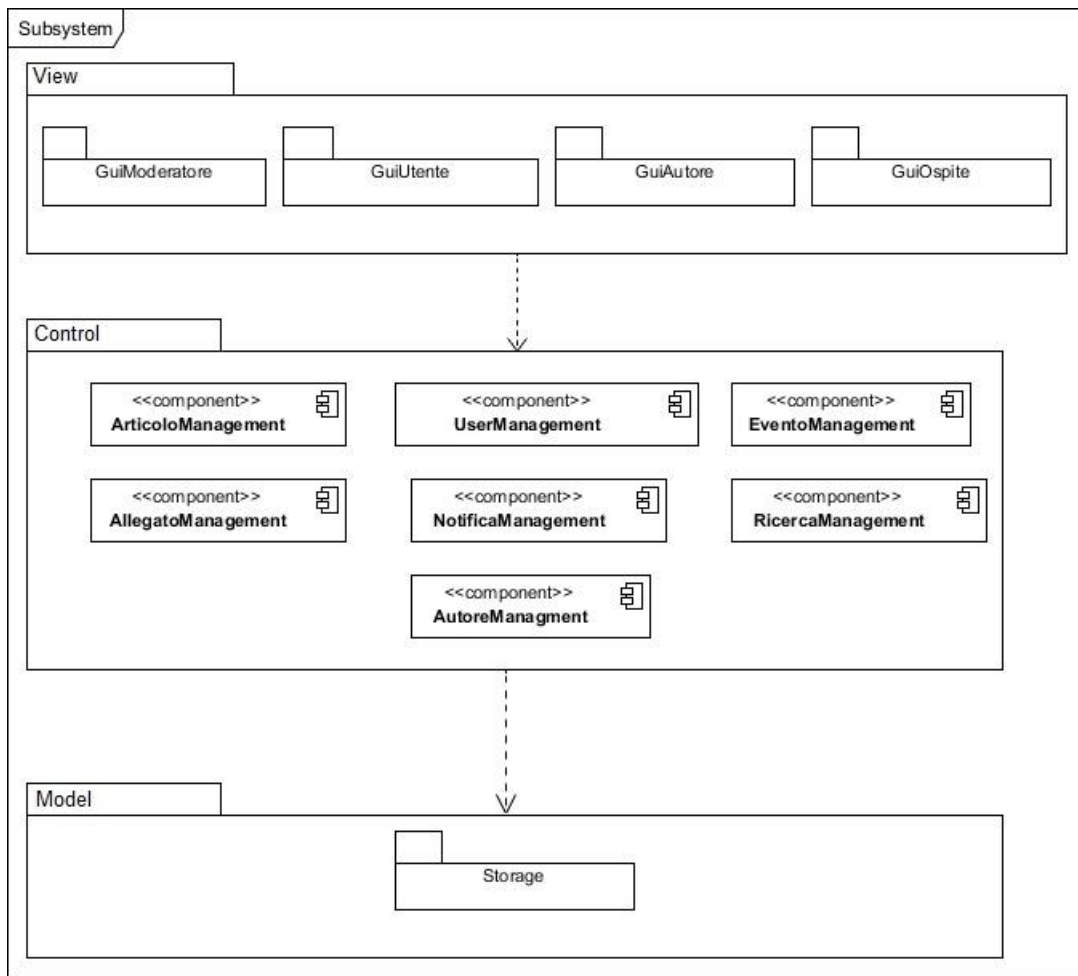
3.2.1 Decomposizione in Layer

La decomposizione prevista per il sistema è quella basata su tre livelli. Quindi abbiamo tre livelli che si occupano di gestire diverse funzionalità del sistema:

- Model: che si occupa della gestione e dell'accesso ai dati
- Control: gestisce la sequenza di interazioni con l'utente
- View: si occupa della parte di visualizzazione delle informazioni della piattaforma

3.2.2 Decomposizione in sottosistemi.

Il sistema è stato suddiviso nei seguenti sottosistemi:

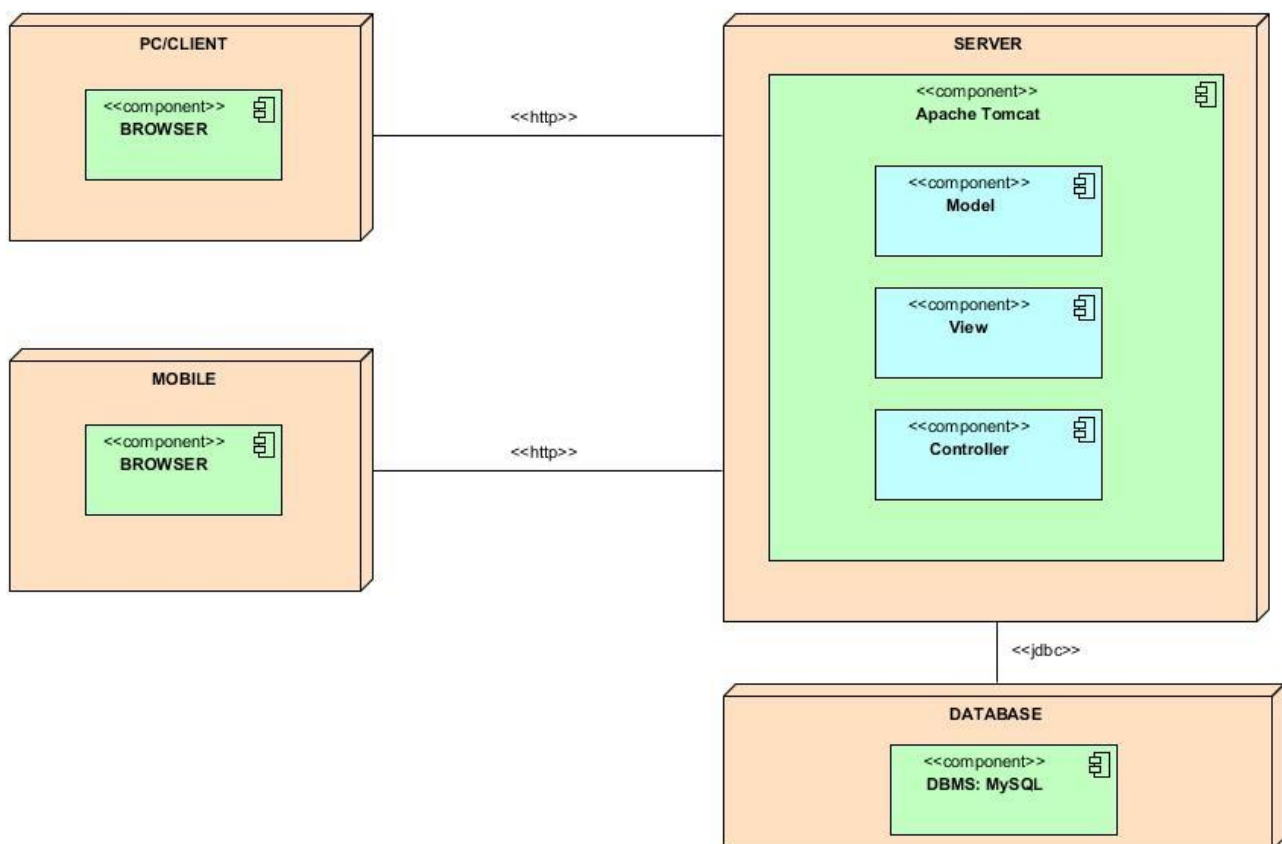


- Il livello View prevede la gestione di quattro sottosistemi
 - o GuiUtente
 - o GuiAutore
 - o GuiModeratore
 - o GuiVisitatore
- Il livello Control prevede la gestione di 7 sottosistemi:
 - o ArticoloManagement:
 - Fornisce una serie di funzionalità per gestire gli articoli
 - o UserManagerment:
 - Fornisce una serie di funzionalità per la gestione delle informazioni utente
 - o EventoManagement:
 - Fornisce una serie di funzionalità per la gestione di eventi
 - o AllegatoManagement:
 - Fornisce una serie di funzionalità per la gestione degli allegati
 - o RicercaManagement:
 - Fornisce una serie di funzionalità per la ricerca

- NotificaManagement
 - Fornisce una serie di funzionalità per gestire le notifiche
- AutoreManagement
 - Fornisce una serie di funzionali per la gestione delle interazioni con l'autore
- il livello Model che prevedere la gestione di un singolo sottosistema
 - Storage

3.3 Hardware/software mapping

Si è scelto di utilizzare una struttura hardware costituita da un server (unico) che risponderà ai servizi richiesti dal client. Il client è un qualsiasi dispositivo dotato di un web browser, che supporta le funzionalità di base di JavaScript, e di una connessione ad Internet che verranno utilizzati per connettersi alla piattaforma. Per il server si è deciso di utilizzare una macchina che include un web server in particolare Apache Tomcat e un DBMS in questo caso MySql per gestire sia la logica che la persistenza dei dati. Il client e il server saranno connessi tramite una rete che utilizzerà il protocollo TCP/IP attraverso cui verranno inoltrate le richieste di un servizio. Il web server comunicherà con il DBMS tramite JDBC.



3.4 Persistent data management

3.4.1 Selecting a storage management strategy

Per memorizzare i dati si è scelto un database relazione che consente la gestione della concorrenza, il controllo degli accessi e il ripristino degli accessi anomali. Inoltre vista la quantità di dati che continuerà ad aumentare visto lo scopo della piattaforma un database relazionale risulta essere scalabile e quindi ideale per grandi quantità di dati. Di conseguenza questo ci permetterà di avere accessi efficienti, tempi di risposta brevi e un'elevata quantità di spazio di archiviazione.

Con scelta unanime, abbiamo optato per l'utilizzo di database MySQL. A differenza dei suoi competitors, MySQL offre servizi altamente ottimizzati per l'utilizzo in grandi sistemi, con uno sguardo sempre attento alle ottimizzazioni circa l'indicizzazione e la richiesta dei dati. Nello specifico, abbiamo trovato adatto al nostro sistema la possibilità di utilizzare:

- Un'interfaccia grafica completa (detta Workbench) per modificare il database on-the-fly;
- Un potente sistema di indicizzazione dei dati che permette di scrivere e prelevare dal database in maniera rapida ed efficiente;
- Una libreria (JDBC-Connector) che ha il pieno supporto del linguaggio che utilizzeremo lato server (Java);
- Un sistema per la gestione dei vincoli referenziali chiara ed efficiente.

3.4.2 Identifying persistent objects

Si è scelto di rendere persistenti all'interno del sistema i seguenti dati:

- Notifica:
 - Viene creata una nuova notifica all'interno del DB quando il moderatore termina la moderazione. Utilizzata per informare l'autore sull'esito della pubblicazione. La chiave primaria per la notifica è un id univoco auto incrementale.
- Autore
 - Utente del sito che può scrivere modificare articoli, ha la possibilità di organizzare eventi e di comunicare con gli utenti della piattaforma. La chiave primaria per autore è Email.
- Messaggio
 - Creato all'interno del DB quando un'utente della piattaforma decide di mettersi in contatto con un autore. Contiene le informazioni relative al messaggio. La chiave primaria di messaggio è un id univoco auto incrementale
- Risposta
 - Creato all'interno del DB quando un autore della piattaforma decide di mettersi in contatto con un utente. Contiene le informazioni relative al messaggio. La chiave primaria di messaggio è un id univoco auto incrementale
 -
- Utente

- Viene creata all'interno del DB dopo che un utente ha effettuato la fase di registrazione. Un utente può svolgere molte delle attività previste dalla piattaforma che vanno dal potersi autenticare a recensire e commentare articolo. La chiave primaria di utente è Email.
- Evento
 - Nel momento in cui l'autore decide di organizzare un evento viene creata una nuova istanza di evento nel DB. L'evento contiene le informazioni su data, luogo e nome dell'evento. La chiave primaria per evento è rappresentata dalla quadrupla luogo, nome e data e un id auto incrementale
- Commento
 - Ogni utente può lasciare un proprio commento ad un articolo. La chiave primaria di commento è rappresentata dall'identificativo dell'utente che lascia il commento e l'identificativo dell'articolo.
- Articolo
 - Viene creata una nuova istanza di articolo all'interno del DB quando il moderatore termina la fase di moderazione e approva l'articolo. Viene utilizzato per tenere traccia delle informazioni scritte da autori. La chiave primaria di Articolo è un id univoco auto incrementale
- Moderatore
 - Rappresenta la persona esperta in un determinato ambito che si occupa della fase di moderazione degli articoli inerenti alla propria categoria di moderazione. La chiave primaria di Moderatore è Email.
- Allegato
 - Viene utilizzata per rappresentare le informazioni aggiuntive (dimostrazioni di ciò che viene specificato nell'articolo) necessarie per l'articolo. Viene creata una nuova istanza all'interno del DB quando l'autore allega il file all'articolo che intende pubblicare. La chiave primaria di allegato è file.
- Rating
 - Rating rappresenta la valutazione da 1 a 5 lasciata da un utente ad un articolo. La chiave primaria di rating è rappresentata dall'identificativo dell'utente che lascia il commento e l'identificativo dell'articolo.

3.5 Access control and security

Attori/Oggetti	Articolo	Allegato	Evento	Rating	Commento	Notifica	Messaggio
Autore	✓Richiesta di pubblicazione articolo. ✓Visualizza Articolo ✓Modifica Articolo	✓Carica Allegato ✓Scarica Allegato ✓Visualizza Allegato ✓Cancella Allegato	✓Organizza Evento ✓Cancella Evento ✓Modifica Evento	✓Visualizza Rating	✓Visualizza Commenti	✓ricevi notifica ✓visualizza notifica	✓invia messaggio ✓ricevi messaggio
Utente	✓Visualizza Articolo	✓Scarica Allegato ✓Visualizza Allegato	✓Visualizza Evento	✓Visualizza Rating ✓Rating	✓Commenta Articolo ✓Visualizza Commenti		✓invia messaggio ✓ricevi messaggio
Moderatore	✓Visualizza Articolo ✓Moderazione Articolo	✓Scarica Allegato ✓Visualizza Allegato				✓invia notifica	

3.6 Global software control

Il flusso del sistema InfoBlog fornisce delle funzionalità che richiedono continua interazione da parte degli utenti, per cui il controllo del flusso globale del sistema è di tipo event-driven, dunque guidato dagli eventi. Il sistema è caratterizzato da un sito web accessibile da un browser e da un webServer, attivo 24h, che deve provvedere a gestire gli accessi concorrenti da parte di utenti. Quando un utente fa il log in e sottomette i propri dati, vi è un accesso al DB per il controllo dell'esistenza dell'utente. Dopo la conferma, l'utente può accedere a diverse operazioni messe a disposizione dal sistema. Ogni operazione è indipendente dalle altre ed è attivabile dalla pressione di un bottone (onClick). Il controllo del flusso viene gestito da Java Servlets che, interagendo con il client, svolgono varie operazioni. Il server smista ogni nuova richiesta alla servlet adeguata, inoltrando poi la risposta al client sotto forma di JSP.

3.7 Boundary condition

Le boundary condition riguardano l'accensione e lo spegnimento del sistema per il solo lato Server.

Avvio del sistema

Per l'avvio del sistema è necessario che l'amministratore avvia prima i servizi del database e poi del web server mediante l'apposita shell. Il sistema dopo essersi avviato presenta risulta raggiungibile dai client

Shutdown server

Prima della disattivazione del sistema verranno disconnessi tutti i client connessi al sistema. Per non incorrere in problemi basta spegnere il server tramite riga di comando. Non vi saranno problemi circa la gestione dei database poiché è gestito dal DBMS.

Eccezioni

In caso di errori di tipo hardware la persistenza dei dati è garantita, così come l'integrità del sistema di memorizzazione scelto in quanto i dati vengono gestiti dal DBMS. In caso di malfunzionamenti del sistema che possono essere dovuti alla perdita di connessione con qualsiasi

componente che va dal client al server al database si è scelto di gestire questo errore mediante ripetuti tentativi di riconnessione. Tuttavia, se questa procedura non risolve il problema sarà compito dell'amministratore utilizzare le giuste procedure per risolvere il problema. In presenza di un errore software che può essere dovuto ad un errore in fase di implementazione di una componente, l'amministratore provvederà ad eseguire i dovuti controlli per risolvere il problema.

ID	UC_01
Use case name	Startup server
Participating actor	Amministratore
Entry condition	L'amministratore è connesso tramite una shell al server che ospita il web server
Flow event	<ol style="list-style-type: none"> 1. L'amministratore avvia inizialmente il servizio di Database 2. Se il Database server era stato spento correttamente allora viene avviato il servizio di esecuzione. Nel caso in cui il Database Server era stato spento in modo anomalo, viene eseguito il check di integrità sui dati, quindi vengono ripristinati i dati corrotti. 3. L'amministratore inserisce a questo punto sulla shell il comando <code>"/startup.sh"</code> 4. Il server è avviato correttamente
Exit condition	Il sever è avviato correttamente e rimane in attesa di connessioni

ID	UC_02
Use case name	Shutdown server
Participating actor	Amministratore
Entry condition	L'amministratore è connesso tramite una shell al server che ospita il web server
Flow event	<ol style="list-style-type: none"> 1. L'amministratore interrompe inizialmente il servizio di Database 2. L'amministratore inserisce a questo punto sulla shell il comando <code>"/shutdown.sh"</code> 3. Il server è arrestato correttamente
Exit condition	Il sever e di conseguenza la piattaforma non è raggiungibile

4. Subsystem services

Servizi forniti da ArticoloManagement	
moderazioneArticolo	Consente la moderazione degli articoli che devono essere pubblicati
rifiutoArticolo	Consente agli autori di rifiutare gli articoli
richiestaPubblicazioneArticolo	Consente agli autori di effettuare la richiesta di pubblicazione di un articolo
showArticle	Consente la visualizzazione di un articolo all'utenza
insertComment	Consente all'utenza di fornire un commento all'articolo
modificaArticolo	Consente all'autore di modificare l'articolo
Rating	Consente agli utenti di poter valutare un articolo mediante una votazione
visualizzaCommenti	Consente all'utenza di poter visualizzare i commenti di un articolo
visualizzaRating	Consente di visualizzare il rating di un articolo
Esplora	Consente di visualizzare gli articoli recentemente pubblicati

Servizi forniti da AllegatoManagement	
CaricaAllegato	Consente all'autore di poter caricare degli allegati
scaricaAllegato	Consente all'utenza di poter scaricare gli allegati presenti all'interno delle pagine
cancellaAllegato	Consente all'autore di cancellare gli allegati
visualizzaAllegato	Consente di visualizzare gli allegati di un articolo

Servizi forniti da EventoManagement	
insertEvento	Consente all'autore di poter organizzare un evento
cancellaEvento	Consente all'autore di poter cancellare gli eventi organizzare
showEvento	Consente all'utente di visualizzare le informazioni dell'evento
modificaEvento	Consente all'autore di poter modificare le informazioni di un evento

Servizi forniti da UserManagement	
Login	Consente all'utenza di effettuare il login
Registrazione	Consente all'utenza di registrarsi alla piattaforma
ModificaDatiPersonali	Consente all'utenza di poter modificare i propri dati
Logout	Consente all'utenza di poter uscire dalla piattaforma
visualizzaDatiPersonali	Consente all'utenza di poter visualizzare i propri dati personali

Servizi forniti da RicercaManagement	
ricercaAutore	Consente all'utenza di effettuare la ricerca degli autori della piattaforma
ricercaArticolo	Consente all'utenza di effettuare la ricerca degli articoli della piattaforma

Servizi forniti da NotificaManagement	
showNotifica	consente di poter visualizzare le notifiche
InviaNotifica	Consente di inviare un notifica per informare sull'esito di una pubblicazione o richiesta di pubblicazione

Servizi forniti da AutoreManagement	
contattaAutore	Consente all'utente di poter comunicare con l'autore
seguiAutore	Consente all'utente di poter seguire un determinato autore
visualizzaPaginaAutore	Consente all'utenza di poter visualizzare le informazioni relative all'autore

Servizi forniti da store a ArticoloManagement	
insertArticolo	Consente di memorizzare le informazioni relativa all'articolo
insertCommento	Consente di memorizzare le informazioni relative ai commenti dell'articolo
insertRating	Consente di memorizzare le informazioni sul rating dell'articolo
UpdateArticolo	Consente di aggiornare le informazioni dell'articolo

Servizi forniti da store a AllegatoManagement	
insertAllegato	Consente di memorizzare le informazioni relativa agli allegati

Servizi forniti da store a EventoManagement	
insertEvento	Consente di memorizzare le informazioni relativa agli eventi
updateEvento	Consente di aggiornare le informazioni relative ad un determinato evento

Servizi forniti da store a UserManagement	
insertUtente	Consente di memorizzare le informazioni relative agli utenti
updateUtente	Consente di aggiornare le informazioni relative ad un determinato utente

Servizi forniti da store a NotificaManagement	
insertNotifica	Consente di memorizzare le informazioni relative alla notifica

Servizi forniti da store a AutoreManagement	
insertMessaggio	Consente di memorizzare le informazioni relative relative al messaggio dell'utente
insertRisposta	Consente di memorizzare le informazioni relative relative alla risposta dell'autore