



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

SDD

System Design Document

Progetto

STORYTELLING

Riferimento	
Versione	0.4
Data	24/10/2021
Destinatario	Prof. Gravino
Presentato da	Alessandro Marigliano, Antonio Scotellaro, Emmanuele Virginio Coppola, Muriel Rossi.
Approvato da	



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

Revision History

Data	Versione	Descrizione	Autori
19/11/2021	0.1	Prima Stesura	Emmanuele Virginio Coppola, Antonio Scotellaro, Alessandro Marigliano, Muriel Rossi
9/12/2021	0.2	Revisione documento e aggiunta di diagrammi	Antonio Scotellaro, Emmanuele Virginio Coppola
10/12/2021	0.3	Aggiunta rank design goals, Use Cases Condizioni Limite, Dizionario dei dati e servizi dei sottosistemi	Antonio Scotellaro, Alessandro Marigliano
12/12/2021	0.4	Revisione Class Diagram e dizionario dei dati, aggiunta matrice degli accessi e revisione totale del documento	Emmanuele Virgino Coppola, Antonio Scotellaro, Alessandro Marigliano



Sommario

1 Introduzione.....	3
1.2 Design Goals.....	3
1.2.1 Trade Offs.....	5
1.3 Definizioni, acronimi e abbreviazioni.....	5
1.4. Riferimenti.....	6
1.5. Panoramica del documento.....	6
3 Architettura del sistema Proposto.....	7
3.1 Panoramica.....	7
3.2 Decomposizione in sottosistemi.....	8
3.3 Mapping hardware/software.....	9
3.4 Gestione dei dati persistenti.....	10
3.5. Controllo degli accessi e sicurezza.....	13
3.6. Controllo flusso globale del sistema.....	13
3.7. Condizioni Limite.....	13
4 Servizi dei Sottosistemi.....	16

1 Introduzione

1.2 Design Goals

ID	Categoria	Nome	Descrizione	Rank
DG_1	Criteri di Utente finale:	Usabilità	Il Sistema sarà di facile utilizzo ed intuitivo dal punto di vista end user.	4
DG_2	Criteri di Affidabilità:	Attendibilità	Il Sistema garantirà l'attendibilità, intesa come corretta gestione delle funzionalità.	5



DG_3	Criteri di Affidabilità:	Disponibilità	Il Sistema dovrà essere permanentemente fruibile dagli utenti.	6
DG_4	Criteri di Affidabilità:	Sicurezza	Ogni Utente è dotato di email e password con cui poter effettuare l'accesso al Sistema.	3
DG_5	Criteri di Affidabilità:	Tolleranza ai guasti	Si eviterà la perdita dei dati e il fallimento di operazioni.	7
DG_6	Criteri di Performance:	Tempi di risposta	Il prodotto Software deve consentire una navigazione rapida ai vari Utenti, quindi, tempi di risposta minimi nello svolgimento delle funzionalità da esso offerte.	8
DG_7	Criteri di Performance:	Memoria	La memoria necessaria al funzionamento del sistema dipende dalla memoria necessaria per il mantenimento del database e del Sistema Operativo	11
DG_8	Criteri di Costo:	Costo di sviluppo	È stimato un costo complessivo di 400 ore per la progettazione e lo sviluppo del sistema.	1
DG_9	Criteri di Manutenibilità:	Adattabilità	Il Sistema può essere adattato solo in base alla tipologia di	9
DG_10	Criteri di Manutenibilità:	Modificabilità	Il Sistema dovrà essere facilmente modificabile.	10



DG_11	Criteri di Manutenibilità:	Estendibilità	I Sistema potrà essere esteso con altre funzionalità.	12
DG_12	Criteri di Manutenibilità:	Tracciabilità	I requisiti del Sistema sono facilmente tracciabili, grazie all'utilizzo di una matrice di tracciabilità.	2

1.2.1 Trade Offs

- **Attendibilità vs Tempi di risposta**

Il Sistema dovrà garantire l'attendibilità, preferendola al tempo di risposta nel caso in cui non sia possibile garantirle entrambe.

- **Disponibilità vs Tolleranza ai guasti**

Il Sistema deve poter essere disponibile all'utente anche nel caso di guasti, che verranno ovviamente segnalati con messaggi di errore, ma permettendo di continuare ad usufruire delle altre funzionalità.

- **Memoria vs Estendibilità**

Il Sistema sarà implementato in modo da preferire l'estendibilità rispetto alla memoria utilizzata, in modo da poter implementare nuove funzionalità, anche se a discapito della memoria.

1.3 Definizioni, acronimi e abbreviazioni

- RAD: Requirements Analysis Document.
- Tirocinio: Tirocinio curriculare esterno.
- UC: Use Case.
- RF: Requisito funzionale.
- NRF: Requisito non funzionale.
- DB: DataBase;



- DBMS: DataBase Management System
- SDD: System Design Document
- MySQL: È un database Open Source basato sul linguaggio SQL, composto da un client a riga di comando e un Server.
- SQL: Structured Query Language; linguaggio standardizzato per database basati sul modello relazionale (RDBMS) progettato per: creare e modificare schemi di database.
- Utente: Un qualsiasi utilizzatore della piattaforma.
- Storia: Breve testo scritto.
- Post: Insieme complessivo della Storia e dei relativi commenti.
- Commento: Breve testo scritto come opinione ad una Storia.
- Reazione: Oggetto per rappresentare il gradimento di una storia.

1.4. Riferimenti

- Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (2nd edition), Prentice-Hall, 2003
- Ian Sommerville, Software Engineering, Addison Wesley
- Statement of work

1.5. Panoramica del documento

Introduzione: Composta da: l'obiettivo del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni allo scopo di chiarire concetti e termini vaghi.

Architettura Sistema Proposto: E' descritta l'architettura del sistema implementato, dove sarà trattata la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

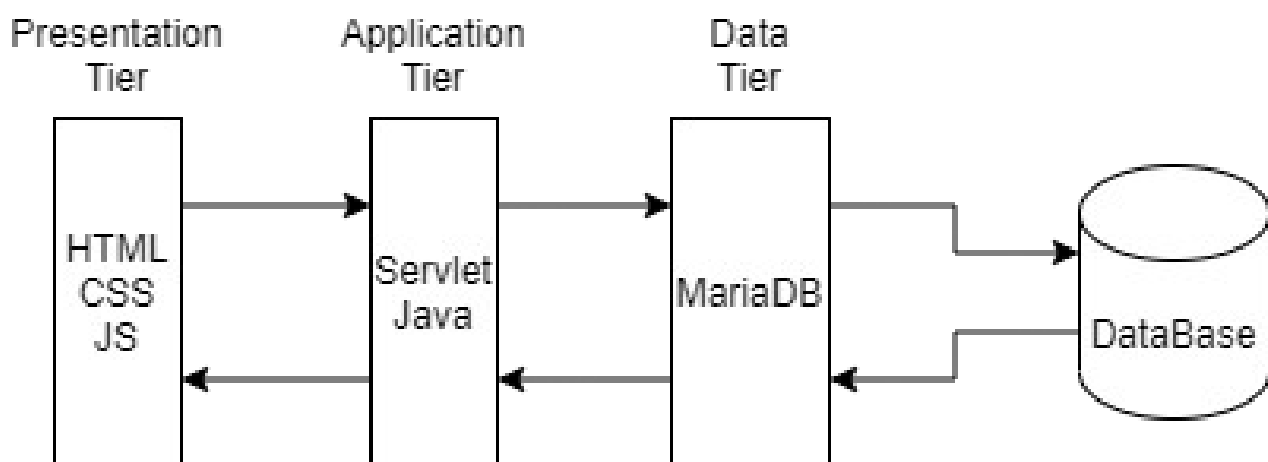
Servizi dei Sottosistemi: Vengono presentati i servizi dei sottosistemi.

Glossario: Raccolta di vocaboli meno comuni utilizzati nella stesura del documento.

3 Architettura del sistema Proposto

3.1 Panoramica

Il pattern architetturale scelto è il Three-Tier Architecture suddividendo il sistema in tre Tier cioè in tre sottosistemi indipendenti che possono essere sviluppati in contemporanea e messi in funzione su 3 macchine separate con il minimo setup.



I 3 Tier sono:

- **Presentation tier:** si occupa dell'interfaccia dell'utente finale mostrandogli informazioni e collezionare input dal suddetto. In questo sistema il tier sarà gestito dal web-browser usando HTML, CSS e JavaScript come linguaggi per gestirne la logica e la presentazione delle informazioni, HTTP per gli scambi di informazioni tra il presentation e l'application tier.
- **Application tier:** definito anche come tier logico o tier intermedio, è il cuore del sistema, elabora le informazioni raccolte presentation tier, a volte confrontandole con le informazioni del data tier e applica la logica di business manipolando il data tier se necessario. In questo sistema l'application tier è gestito da Servlet scritte in Java.



- **Data tier:** a volte definito come database tier, tier di accesso dati o back-end, è dove le informazioni sono processate, archiviate e gestite. In questo sistema il data tier è gestito da un sistema di database relazionale MariaDB.

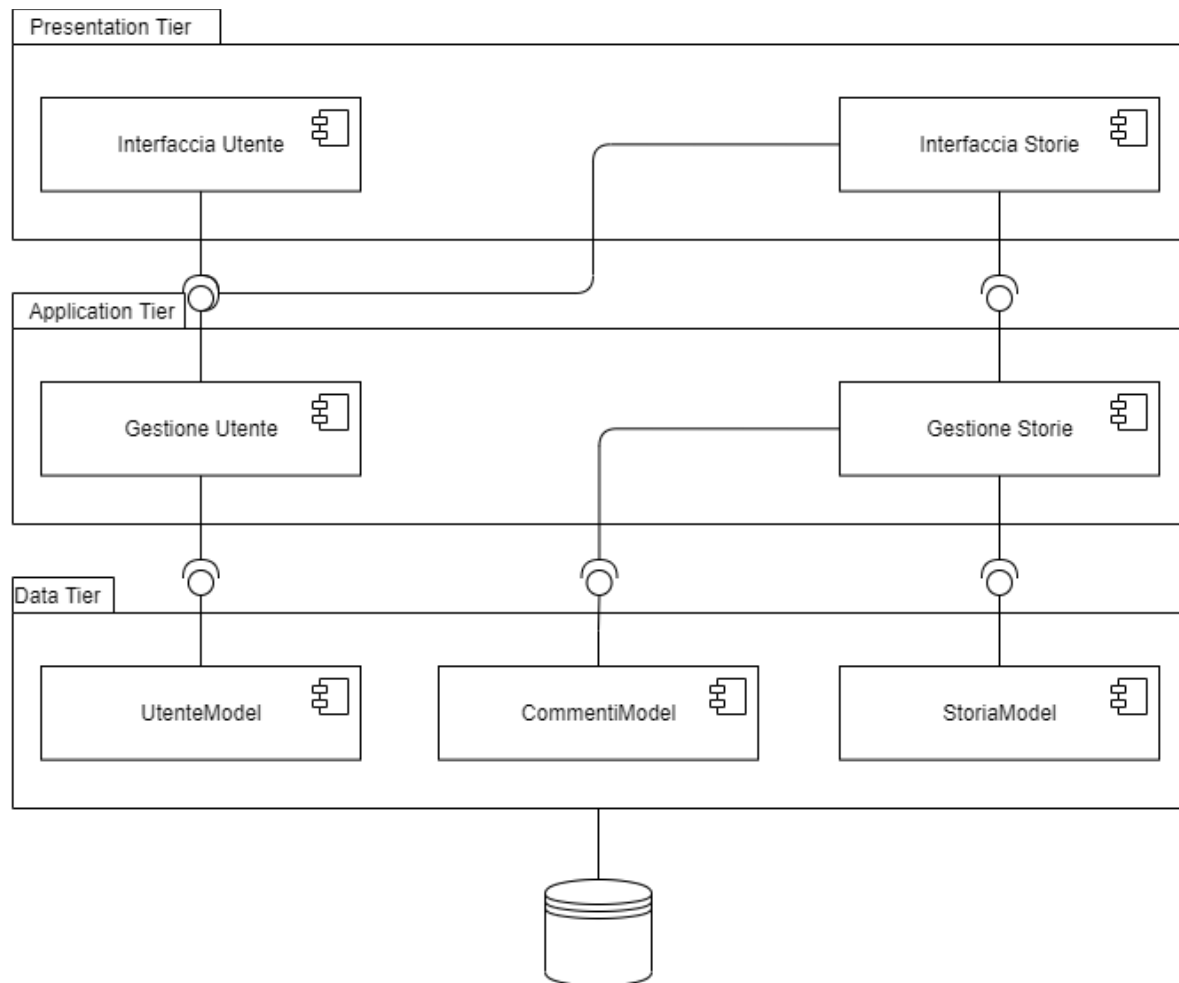
3.2 Decomposizione in sottosistemi

Il sistema è diviso in due sottosistemi principali:

Il Sottosistema **Gestione Account**, che si occupa della gestione delle informazioni sensibili dell'utente garantendone la sicurezza e in più se necessario eliminare un'utente se esso si rileva un soggetto nocivo per la piattaforma.

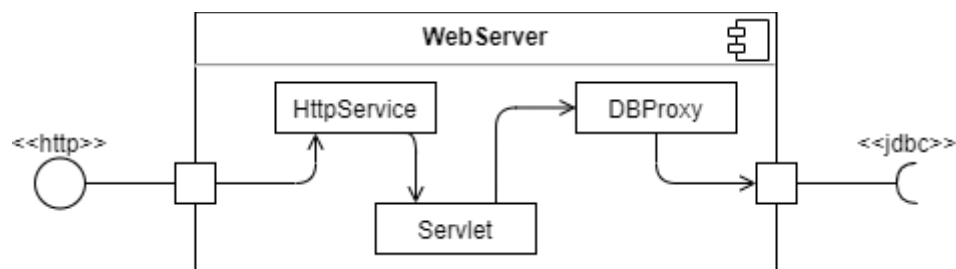
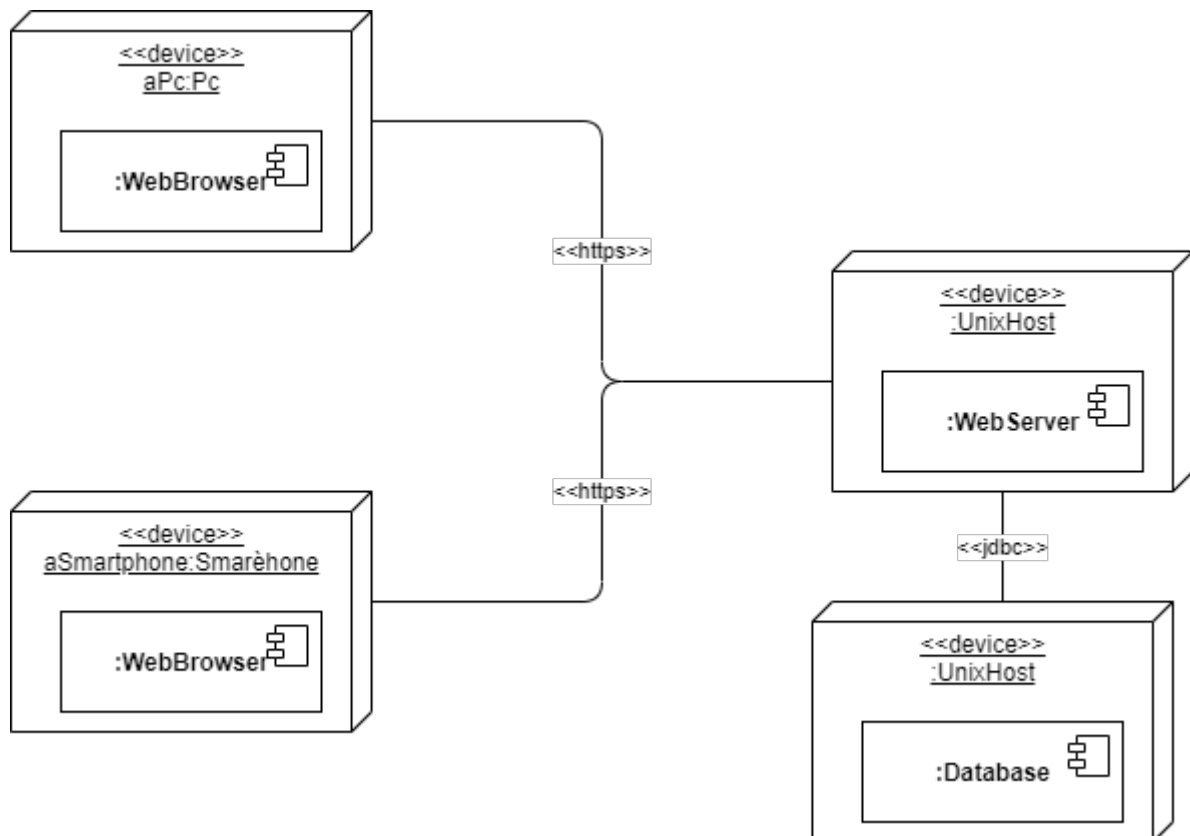
Le operazioni che gestiscono queste funzionalità sono Registrazione, Login, Logout, Elimina Account ed Eliminazione Spammer.

Il Sottosistema **Gestione Bacheca**, che si occupa di tutte le funzionalità in merito alla gestione della visualizzazione delle storie ,il loro inserimento e tutte le interazioni con esse, in particolare delle operazioni di: Pubblicazione Storia, Inserimento Reazione, Inserimento Commento, Visualizzazione Post e Visualizzazione Bacheca.



3.3 Mapping hardware/software

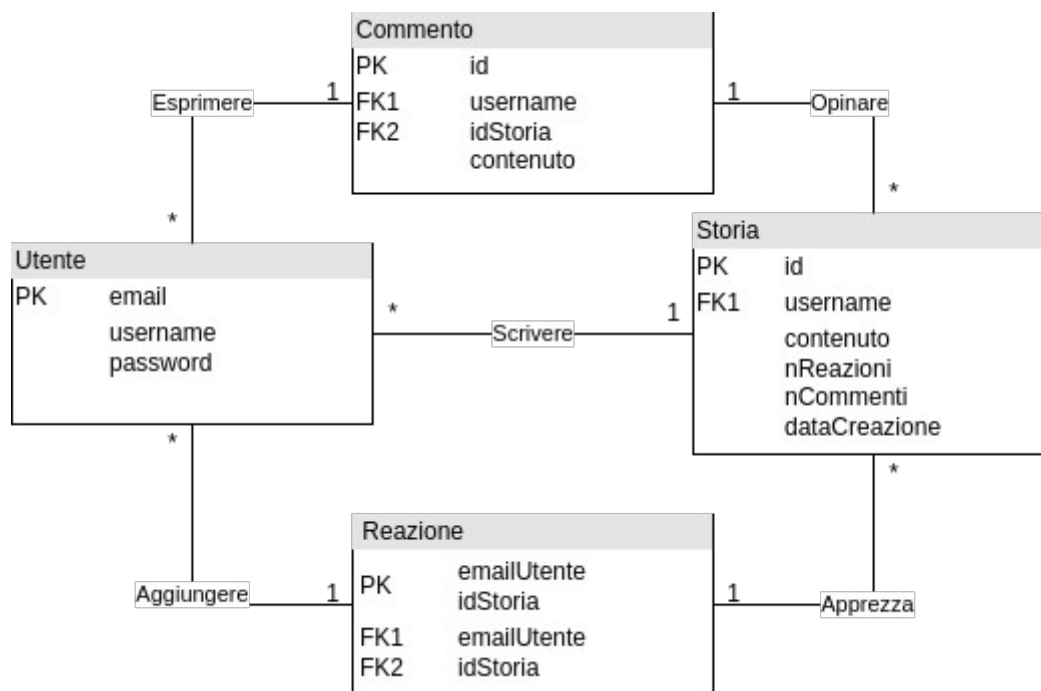
Il sistema da sviluppare utilizzerà una struttura hardware composta da un client ed un server dove il server si occupa di tutta la logica di business e la gestione dei dati persistenti mentre il client fornisce un'interfaccia grafica che permette all'utente di interagire con le funzionalità del sistema. I requisiti del client sono una connessione ad Internet ed un web browser compatibile con tecnologie CSS3 e HTML5 mentre i requisiti del server sono una connessione ad Internet, la capacità di immagazzinare una quantità enorme di dati e un DBMS che permetta la comunicazione con i client.



3.4 Gestione dei dati persistenti

Per la gestione dei dati persistenti abbiamo scelto di utilizzare un Database relazionale che ci permette di accedere alle informazioni in modo rapido ed efficiente e inoltre di immagazzinare una grande quantità di dati. Come schema concettuale per mantenere una coerenza con quanto detto precedentemente nelle scorse fasi di analisi abbiamo scelto di utilizzare come tipologia di rappresentazione il Class Diagram.

Da notare che la password all'interno del database non è la password in chiaro ma un hash della password relativa all'utente in base all'algoritmo sha512



Utente

NAME	TYPE	ALTRI VINCOLI	KEY
email	Varchar(300)	NOT NULL	PRIMARY
password	Varchar(64)	NOT NULL	
username	Varchar(20)	NOT NULL, UNIQUE	

Storia

NAME	TYPE	ALTRI VINCOLI	KEY
id	Int	NOT NULL, AUTO-INCREMENT	PRIMARY
username	Varchar(20)	NOT NULL	FOREIGN
contenuto	Varchar(512)	NOT NULL	
nReazioni	int	NOT NULL	



nCommenti	int	NOT NULL	
dataCreazione	Date	NOT NULL	

Reazione

NAME	TYPE	ALTRI VINCOLI	KEY
emailUtente	Varchar(300)	NOT NULL	PRIMARY FOREIGN
idStoria	Int	NOT NULL	PRIMARY FOREIGN

Commento

NAME	TYPE	ALTRI VINCOLI	KEY
id	Int	NOT NULL, AUTO-INCREMENT	PRIMARY
idStoria	Int	NOT NULL	FOREIGN
username	Varchar(300)	NOT NULL	FOREIGN
contenuto	Varchar(128)	NOT NULL	



3.5. Controllo degli accessi e sicurezza

Attori	Utente	Modulo FIA
Oggetti		
Account	Login, Logout, Registrazione, Eliminazione Account	Login, Logout, Eliminazione Account
Bacheca	Pubblicazione, Storia, Inserimento Reazione, Inserimento Commento, Visualizzazione Post, Visualizzazione Bacheca	Visualizzazione Post, Visualizzazione bacheca

3.6. Controllo flusso globale del sistema

Il sistema StoryTelling è un sistema le cui funzionalità vengono avviate in base a comandi impartiti dall'utente (o bot di eliminazione spammers) tramite l'uso di interfacce grafiche. Siccome il sistema oltre ad avere queste caratteristiche è anche una web-application il flusso del sistema è di tipo event-driven, ovvero guidato dagli eventi.

3.7. Condizioni Limite

Identificativo UC_BC_1	StartUp	Data	10/12/21
		Vers.	0.00.001
		Autore	Antonio Scotellaro
Descrizione	Lo use case permette l'avvio del sistema unix e il successivo avvio del server su cui si basa il funzionamento della piattaforma		
Attore Principale	Sistema		
Attori secondari	NA		



Entry Condition		Il sistema è spento
Exit condition On success		Il sistema viene avviato correttamente
Exit condition On failure		Il sistema non viene avviato
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Sistema:	Avvio del sistema e richiesta accesso al server
2	Server:	Accetta la richiesta e rende disponibili le sue funzionalità
3	Server:	Manda messaggio di conferma avvenuta connessione
I SCENARIO/FLUSSO DI EVENTI DI ERRORE: IL SISTEMA NON RIESCE A CONNETTERSI AL SERVER		
2.1	Sistema:	Il Sistema avvia un timer di 10 secondi e riproverà a connettersi al server allo scadere del tempo.

Identificativo UC_BC_2		Terminazione	Data	10/12/21
			Vers.	0.00.001
			Autore	Antonio Scotellaro
Descrizione		Lo use case definisce lo spegnimento del sistema		
Attore Principale		Sistema		
Attori secondari		NA		
Entry Condition		Il sistema è avviato AND il sistema è connesso al server		
Exit condition On success		Il sistema viene spento		
Exit condition On failure		Il sistema non viene spento		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Sistema:	Ricezione richiesta spegnimento sistema		
2	Sistema:	Controlla ci siano ancora delle connessioni aperte		
3	Sistema:	Termina l'esecuzione del sistema		
I SCENARIO/FLUSSO DI EVENTI DI ERRORE: IL SISTEMA NON RIESCE A TERMINARE L'ESECUZIONE				
2.1	Sistema:	Il Sistema notifica la presenza di connessione ancora in corso		

Identificativo	Fallimento	Data	10/12/21
----------------	------------	------	----------



UC_BC_3		Vers.	0.00.001
		Autore	Antonio Scotellaro
Descrizione	Lo use case definisce il comportamento del sistema in caso di fallimento critico che termini l'operazione del sistema in maniera inaspettata		
Attore Principale	Sistema		
Attori secondari	NA		
Entry Condition	Il sistema viene terminato inaspettatamente		
Exit condition On success	Il sistema viene riavviato con successo		
Exit condition On failure	Il sistema non viene riavviato		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
STESSO FLUSSO DI EVENTI DI UCBC_1			
1	Sistema:	Avvio del sistema e richiesta accesso al server	
2	Server:	Accetta la richiesta e rende disponibili le sue funzionalità	
3	Server:	Manda messaggio di conferma avvenuta connessione	



4 Servizi dei Sottosistemi

Sottosistema Utente

Servizio	Descrizione	Interfaccia
Registrazione Account	Questa funzionalità permette di registrare un proprio account sulla piattaforma	Controller
Login	Questa funzionalità permette di fare l'accesso al proprio account sulla piattaforma	Controller
Logout	Questa funzionalità permette di uscire dal proprio account una volta acceduto alla piattaforma	Controller
Eliminazione Account	Questa funzionalità permette di cancellare il proprio account dalla piattaforma	Controller
Eliminazione Spammer	Questa funzionalità elimina gli utenti risultati non conformi alle linee guida della piattaforma	EliminazioneSpammerController

Sottosistema Bacheca

Servizio	Descrizione	Interfaccia
Visualizza Bacheca	Questa funzionalità permette di visualizzare la bacheca presente sulla piattaforma	Controller
Pubblicazione Storia	Questa funzionalità permette di pubblicare all'interno della bacheca una storia	Controller
Visualizza Post	Questa funzionalità permette di visualizzare una storia con i	Controller



	commenti da essa ricevuti	
Inserisci Commento	Questa funzionalità permette di inserire un commento a una storia pubblicata	Controller
Inserisci Reazione	Questa funzionalità permette di inserire una reazione a una storia pubblicata	Controller

4.Glossario

C

Commento: Quando un'utente esprime un pensiero diretto ad una storia di un'altro utente

N

NRF: Requisito non funzionale.

R

RAD: Il documento dell'analisi dei requisiti è un'attività preliminare allo sviluppo di un Sistema software. Lo scopo principale di tale documento è di definire le funzionalità del Sistema.

Reazione: Un attributo che si lascia ad una storia per coinvolgere l'utente alla piattaforma

RF: Un requisito funzionale è un requisito che definisce una funzione del Sistema identificato durante l'analisi dei requisiti

S

Storia: Contenuto che può essere creato da un'utente a cui si possono associare commenti e reazioni

U

UC: Uno o più Use Case vengono utilizzati durante il RAD per identificare l'interazione tra attore e Sistema.