|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SDD  System Design Document  STORYTELLING   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.1 | | Data | 24/10/2021 | | Destinatario | Prof. Gravino | | Presentato da | Alessandro Marigliano, Antonio Scotellaro, Emmanuele Virginio Coppola, Muriel Rossi. | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |
|  |  |  |  |

Sommario

[RevisionHistory 2](#_Toc89953943)

[1.2. Design Goals 4](#_Toc89953944)

[1.2.1 Trade Offs 6](#_Toc89953945)

[2.0 Architettura del sistema corrente 8](#_Toc89953946)

[3.2. Decomposizione in sottosistemi 8](#_Toc89953947)

[3.3. Mapping hardware/software 10](#_Toc89953948)

[3.4. Gestione dei dati persistenti 10](#_Toc89953949)

[3.5. Controllo degli accessi e sicurezza 10](#_Toc89953950)

[3.6. Controllo flusso globale del sistema 10](#_Toc89953951)

[3.7. Condizioni Limite 11](#_Toc89953952)

[4.0 Servizi dei Sottosistemi 11](#_Toc89953953)

## 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. | 5 |
| DG\_2 | Criteri di Affidabilità: | Attendibilità | Il Sistema garantirà l’attendibilità, intesa come corretta gestione delle funzionalità. | 3 |
| 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. | 1 |
| DG\_5 | Criteri di Affidabilità: | Tolleranza ai guasti | Si eviterà la perdita dei dati e il fallimento di operazioni. | 8 |
| 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. | 2 |
| 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 | 4 |
| DG\_8 | Criteri di Costo: | Costo di sviluppo | È stimato un costo complessivo di 400 ore per la progettazione e lo sviluppo del sistema. | 9 |
| DG\_9 | Criteri di Manutenibilità: | Adattabilità | Il Sistema può essere adattato solo in base alla tipologia di | 12 |
| DG\_10 | Criteri di Manutenibilità: | Modificabilità | Il Sistema dovrà essere facilmente modificabile. | 10 |
| DG\_11 | Criteri di Manutenibilità: | Estendibilità | l Sistema potrà essere esteso con altre funzionalità. | 11 |
| DG\_12 | Criteri di Manutenibilità: | Tracciabilità | I requisiti del Sistema sono facilmente tracciabili, grazie all’utilizzo di una matrice di tracciabilità. | 7 |

### 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 Wesely

• 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.

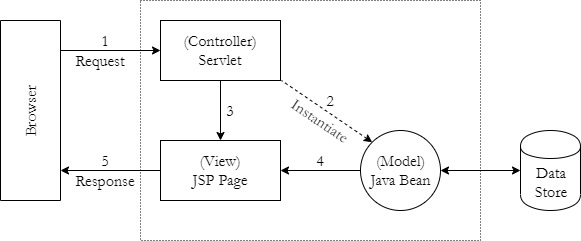
**Glossario**: Raccolta di vocaboli meno comuni utilizzati nella stesura del documento.

# 2.0 Architettura del sistema corrente

**3. Architettura del Sistema proposto**

**3.1.** **Panoramica**

Il pattern architetturale scelto è MVC (Model, View, Control).



**Model**: contiene i metodi di accesso ai dati.

**View**: si occupa di visualizzare i dati all’Utente e gestisce l’interazione fra quest’ultimo e l’infrastruttura sottostante.

**Controller**: riceve i comandi dell’Utente attraverso il View e reagisce eseguendo delle operazioni che possono interessare il Model e che portano generalmente ad un cambiamento di stato del View.

# 3.2. Decomposizione in sottosistemi

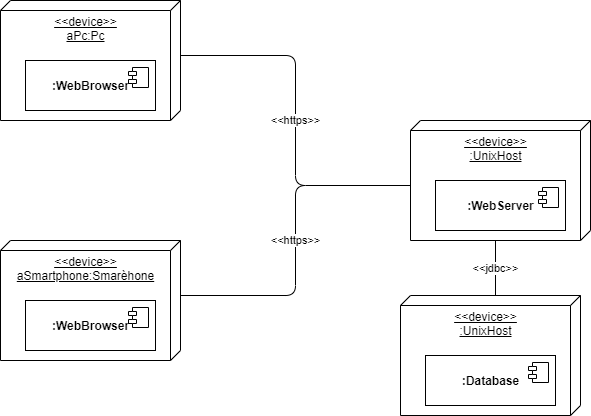
Il sistema è diviso in due sottosistemi principali:

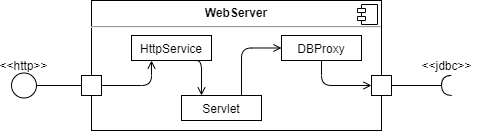
Il Sottosistema ***Gestione Account***, che si occupa della gestione dell’account dell’Utente, in particolare delle operazioni di: Login, Logout, Registrazione ed Eliminazione dal Sistema.

Il Sottosistema ***Gestione Bacheca***, che si occupa di tutte le funzionalità in merito alla gestione della bacheca, in particolare delle operazioni di: Pubblicazione Storia, Inserimento Reazione, Inserimento Commento, Visualizzazione Post e Visualizzazione Bacheca.

# 3.3. Mapping hardware/software

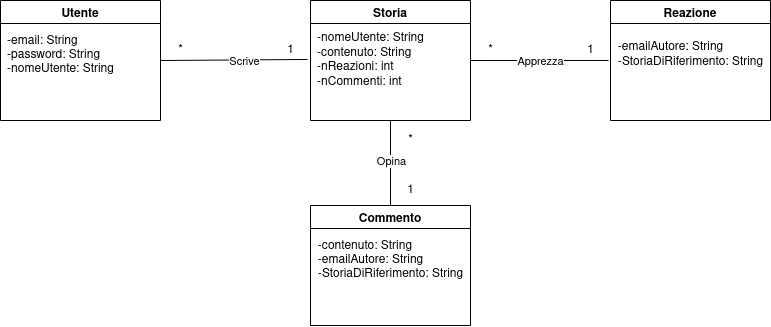
Il sistema da sviluppare utilizzerà una struttura hardware composta da un client ed un server dove il server si occupa di tutta la logica di business e la gestione dei dati persistenti mentre il client fornisce un‘interfaccia grafica che permette all’utente di interagire con le funzionalità del sistema. 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



**Utente**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | TYPE | NULL | KEY |
| Email | String | NOT NULL |  |
| Password | String | NOT NULL |  |
| nomeUtente | String | NOT NULL |  |

**Storia**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | TYPE | NULL | KEY |
| nomeUtente | String | NOT NULL |  |
| contenuto | String | NOT NULL |  |
| nReazioni | Int | NOT NULL |  |
| nCommenti | Int | NOT NULL |  |

**Reazione**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | TYPE | NULL | KEY |
| emailAutore | String | NOT NULL |  |
| storiaDiRiferimento | String | NOT NULL |  |

**Commento**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | TYPE | NULL | KEY |
| contenuto | String | NOT NULL |  |
| emailAutore | String | NOT NULL |  |
| storiaDiRiferimento | String | NOT NULL |  |

# 3.5. Controllo degli accessi e sicurezza

# 3.6. Controllo flusso globale del sistema

Il flusso del sistema è completamente guidato dalle interazioni dell’utente, ciò significa che il sistema è event-driven, ovvero è guidato dagli eventi.

# 3.7. Condizioni Limite

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificativo**  *UCBC\_1* | | | | *StartUp* | *Data* | *10/12/21* |
| *Vers.* | *0.00.001* |
| *Autore* | Antonio Scotellaro |
| **Descrizione** | | | | L’use case permette l’avvio del sistema | | |
| **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**  *UCBC\_2* | | | | *Terminazione* | *Data* | *10/12/21* |
| *Vers.* | *0.00.001* |
| *Autore* | Antonio Scotellaro |
| **Descrizione** | | | | L’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**  *UCBC\_3* | | | *Fallimento* | *Data* | *10/12/21* |
| *Vers.* | *0.00.001* |
| *Autore* | Antonio Scotellaro |
| **Descrizione** | | | L’use case definisce il comportamento del sistema in caso di fallimento | | |
| **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 | | | |

# 4.0 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 Spammmer | 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 commenti da essa ricevuti | Controller |
| 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 |