**BOOKWORM**

**System Design**

**Document**

**TEAM DEL PROGETTO**

Nisivoccia Giuseppe 051210 5804

|  |  |  |
| --- | --- | --- |
| **Data** | **Cambiamenti** | **Autori** |
| 29/07/2021 | Creazione documento | Nisivoccia Giuseppe |
| 25/09/2021 | Aggiunto Class Diagram in dati persistenti | Nisivoccia Giuseppe |
| 28/09/2021 | Aggiunto Servizi dei sottoinsieme e altro | Nisivoccia Giuseppe |
| 12/10/2021 | Revisione documento | Nisivoccia Giuseppe |

Sommario

[​ 1.INTRODUZIONE 4](#__RefHeading___Toc29567_3038064823)

[​ 1.1 Obiettivi del sistema 4](#__RefHeading___Toc29569_3038064823)

[​ 1.2 Design Goals 4](#__RefHeading___Toc29571_3038064823)

[​ 1.3 Riferimenti 5](#__RefHeading___Toc29573_3038064823)

[​ 1.4 Panoramica 6](#__RefHeading___Toc29575_3038064823)

[​ 2.Architettura del Sistema Proposto 6](#__RefHeading___Toc29577_3038064823)

[​ 2.1 Panoramica 6](#__RefHeading___Toc29579_3038064823)

[​ 2.2 Decomposizione in Sistemi 6](#__RefHeading___Toc29581_3038064823)

[​ 2.3 Mapping HW/SW 8](#__RefHeading___Toc29583_3038064823)

[​ 2.4 Gestione dei dati persistenti 9](#__RefHeading___Toc29587_3038064823)

[​ 2.5 Controllo degli accessi e sicurezza 11](#__RefHeading___Toc16782_2333884176)

[​ 2.6 Controllo del sistema globale 12](#__RefHeading___Toc29589_3038064823)

[​ 3.SERVIZI DEI SOTTOSISTEMI 12](#__RefHeading___Toc485_698247756)

# 1.INTRODUZIONE

## 1.1 Obiettivi del sistema

Questo sistema cerca di fondere le funzionalità dei social network più famosi nel tentativo di creare un sistema di comunicazione adatto al pubblico a cui si riferisce.

Il nome BookWorm è stato scelto in riferimento al suo significato che è quello di topo di biblioteca.

Le figure principali che saranno presenti nel sistema sono 3:

-L’Utente: colui che utilizza e sfrutta il sito per puro divertimento

-Il moderatore: colui che gestisce una sua community personale

L’utente può essere anche moderatore e viceversa a seconda delle azioni

che compiono

## 1.2 Design Goals

I design goals per BookWorm sono:

* **Criteri di Performance**
  + Tempo di risposta
    - Il tempo di risposta per ogni azione dell’utente deve essere non superiore ai 5 secondi nonostante il sistema fosse sottoposto ad un carico elevato
    - Il tempo di reindirizzamento ad una pagina deve essere molto breve, la comunicazione tra il database e il sistema deve essere altrettanto veloce
  + Memoria
    - BookWorm utilizza un database relazionale per memorizzare I dati
    - Il database non dovrebbe influenzare le prestazioni del sistema
* **Criteri di Affidabilità**
  + Affidabilità
    - Gli input inseriti dall’utente non validi saranno segnalati come errori
    - Ogni dato relativo a post,community e credenziali dell’utente saranno salvate in modo sicuro nel database del sistema
  + Disponibilità
    - BookWorm al momento è disponibile solo ad accesso limitato a pochi utenti
    - BookWorm in futuro dovrà poter essere disponibile 24/7 a tutti gli utenti con un server dedicato
  + Security
    - Il sistema è accessibile a tutti gli utenti registrati
    - Gli amministratori invece avranno già degli account di default
  + Robustezza
    - Il sistema sarà in grado di mostrare alert di errori dovuti ad una non corretta immissione dei dati da parte dell’utente
* **Criteri di Costo**
  + Costi di Sviluppo
    - Il Server dedicato al sistema verrà hostato da HostGator al prezzo mensile di 2.64$
* **Criteri di Manutenzione**
  + Estendibilità
    - Utilizzando un linguaggio di markup HTML5 e lo stile CSS il sistema sarà il più flessibile possibile in modo da poter implementare nuove feature facilmente
  + Modificabilità
    - Il codice deve rispettare il manuale del buon programmatore, deve essere leggibile e commentato
  + Portabilità
    - Il sistema necessita esclusivamente di un browser
* **Criteri utenti finali**
  + Usabilità
    - Il sistema deve essere facile da utilizzare

## 1.3 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

## 1.4 Panoramica

Le uniche informazioni che BookWorm riceverà saranno quelle che l’utente, di propria iniziativa inserirà per farsi conoscere dagli altri utenti.

Non tutte le informazioni inserite potranno essere modificate ma le più essenziali sì, per essere sempre aggiornati.

Dovute restrizioni aiuteranno l’utente ad utilizzare e gestire il proprio dispositivo e le proprie informazioni in modo anche più responsabile per evitare possibili disagi.

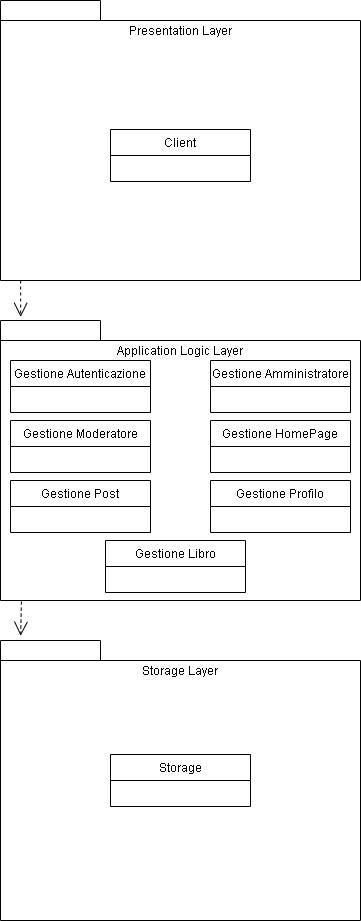
# 2.Architettura del Sistema **Proposto**

## 2.1 Panoramica

BookWorm utilizza lo stile a 3 Strati Client/Server

* Presentation Layer
  + Include le interfacce grafiche e gli oggetti con cui interagisce l’utente
* Application Login Layer
  + Include la logica del sistema
* Storage Layer
  + Include la memorizzazione dei dati tramite DBMS

## 2.2 Decomposizione in Sistemi



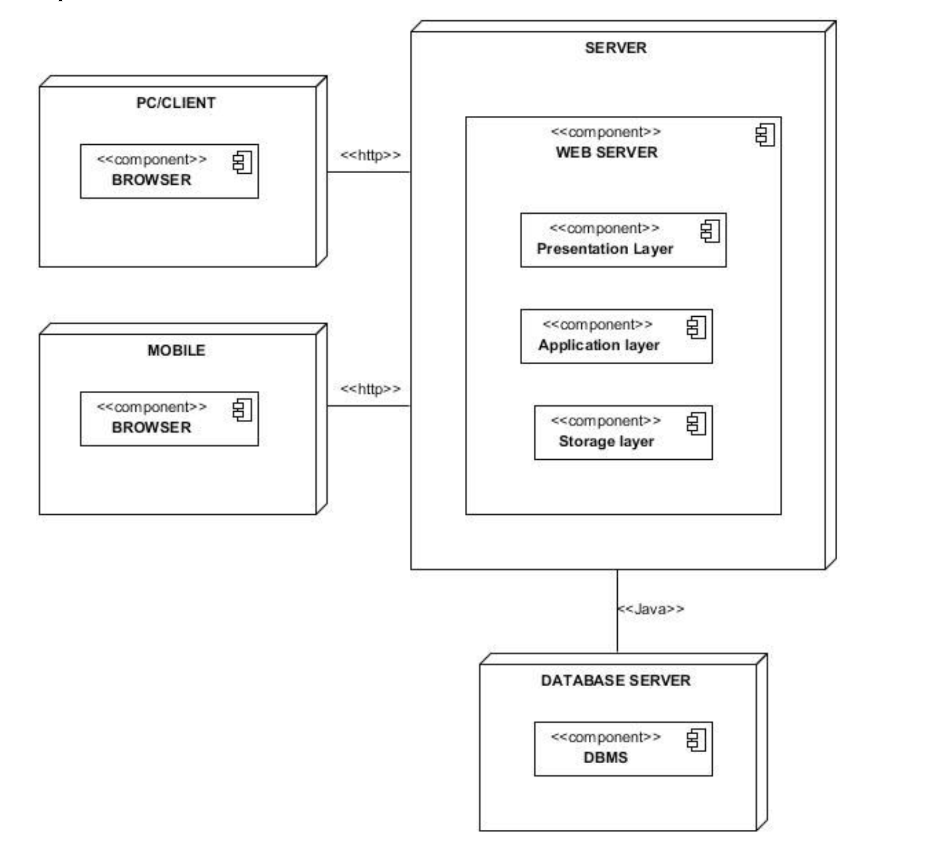
Gestioni del sistema:

* Gestione Autenticazione
  + Serie di funzionalità per l’autenticazione sulla piattaforma
* Gestione Amministratore
  + Serie di funzionalità per la gestione dell’amministratore sulla piattaforma
* Gestione Moderatore
  + Serie di funzionalità per la gestione del moderatore di una Community sulla piattaforma
* Gestione HomePage
  + Serie di funzionalità della HomePage sulla piattaforma
* Gestione Post
  + Serie di funzionalità per la gestione dei Post sulla piattaforma
* Gestione Profilo
  + Serie di funzionalità per la gestione del profilo sulla piattaforma
* Gestione Libro
  + Serie di funzionalità dei Libri sulla piattaforma

## 2.3 Mapping HW/SW

Il sistema avrà un server dove verrà hostato e gli utenti utilizzerano I browser per accedere al sistema

Il protocollo usato per trasferire dati tra client e server sarà HTTP



## 

## 2.4 **Gestione dei dati persistenti**

## 

è stato deciso di utilizzare un DBMS relazionale (MYSQL) per le seguenti motivazioni:

* Accesso ai dati tramite un linguaggio universale (SQL).
* Accesso efficiente ai dati. Un DBMS ha molti modi per ottimizzare l'accesso all'informazione. La base di dati è solitamente memorizzata in memoria secondaria. Un DBMS permette di creare dei file ausiliari (indici) che permettono l'accesso veloce ai dati su disco
* Indipendenza dei dati. Un DBMS mantiene diversi livelli di astrazione dei dati, permettendo di accedere ai dati logici indipendentemente dalla loro rappresentazione fisica.
* Controllo della ridondanza dei dati. Ogni dato logico dovrebbe essere memorizzato in un solo posto nella base di dati. Avere più copie della stessa informazione ha i seguenti svantaggi:
  + Maggior uso di memoria
  + Le modifiche della stessa informazione debbono essere effettuate diverse volte;
  + Ci possono essere fenomeni di inconsistenza dei dati qualora gli aggiornamenti dei dati vengano eseguiti in modo indipendente;
* Atomicità delle operazioni. Un DBMS permette di effettuare sequenze di operazioni in modo atomico. Ciò significa che l'intera sequenza di operazioni viene eseguita con successo oppure nessuna di queste operazioni ha alcun effetto sui dati della base.
* Accesso concorrente ai dati. Un DBMS permette a più utenti di accedere contemporaneamente alla base di dati. Più utenti possono accedere nello stesso istante a dati diversi.
* Privatezza dei dati. Un DBMS permette un accesso protetto ai dati. Utenti diversi possono avere accesso a diverse porzioni della base di dati e possono essere abilitati a diverse operazioni su di esse.
* Affidabilità dei dati. Un DBMS offre dei metodi per salvare copie dei dati (backup) e per ripristinare lo stato della base di dati in caso di guasti software e hardware (recovery).

Per effettuare la ristrutturazione dello schema E-R si è proceduto innanzitutto a non generalizzare l’Utente: Utente-Moderatore-Amministratore.

L’alternativa più adatta alla nostra situazione, sarebbe stata una generalizzazione totalee di mantenere il genitore ed eliminare i figli della generalizzazione.

La scelta è dovuta ai vantaggi che questa soluzione porta in termini di tempi di accesso alle informazioni e migliora la gestione della memoria.

## **2.5 Controllo degli accessi e sicurezza**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Attori** | Utente | Utente  Registrato | Amministratore | Moderatore |
| **Oggetti** |  | | | | |
| Autenticazione |  | * Registrazione * RecuperaPassword * Logout * Login | * AttivazioneAccount * richiediAttivazione | * Registrazione * RecuperaPassword * Logout * Login | * Registrazione * RecuperaPassword * Logout * Login |
| Amministrazione |  |  | * Aggiungi libro * Modifica libro * Rimuovi libro * BanUser * VisualizzaListaLibri * VisualizzaListaUtenti * VisualizzaLista segnalazioni * Visualizza segnalazione * AccettaSegnalazione * RifiutaSegnalazione |  |
| Moderazione |  |  |  | * Modifica foto community * Modifica community * Elimina community * Visualizza Utenti |
| HomePage | * Ricerca | * Ricerca | * Ricerca | * Ricerca |
| Post |  | * Carica post * Modifica post * Inserisci commento * NuovaSegnalazione * RimuoviPost * Like * RemoveLike * VisualizzaLike |  | * RimuoviPost |
| Profilo | * VisualizzaProfilo * ModificaPassword | * Modifica password * Modifica profilo * Modifica immagine * Elimina account * Visualizza profilo * Read Libro * Unread Libro |  | * VisualizzaProfilo * ModificaPassword |
| Libro | * Visualizza libro * Follow community * Unfollow community * Visualizza community * CreaCommunityByBook * VisualizzaCommunity | * Visualizza libro * Follow community * Unfollow community * Visualizza community * CreaCommunityByBook * VisualizzaCommunity |  | * Visualizza libro * Follow community * Unfollow community * Visualizza community * CreaCommunityByBook * VisualizzaCommunity |

## 2.6 **Controllo del sistema globale**

Il sistema è caratterizzato da un sito web accessibile dal browser.

Nel momento in cui un utente effettua il Login, vi è un accesso al Database per controllare l’effettiva esistenza di quest’ultimo.

Avuta la conferma dell’esistenza, l’utente può accedere alle diverse funzionalità messe a disposizione da parte del sistema.

L’utente può richiedere una risorsa attivando una URL. Il browser costruisce una HTTP Request con le informazioni prese dall’URL e viene inviato al server HTTP, in particolare viene catturata da un Controller.

Il Controller riceve l’HTTP Request tradotta in oggetto Java, dopo averlo letto utilizza i dati estratti per eseguire delle operazioni, opzionalmente utilizzando i Model di riferimento dell’oggetto desiderato.

Termina con la realizzazione di una risposta inviata sotto forma di Response ad una View, che avrà il compito di leggere i dati e formattarli in pagina Web.

# 3.SERVIZI DEI SOTTOSISTEMI

* Gestione Autenticazione
  + Registrazione – consente all’utente di registrarsi al sito
  + AttivazioneAccount – consente all’utente di attivare il proprio account
  + RichiediAttivazione - consente all’utente di richiedere la conferma di attivazione
  + RecuperaPassword – consente all’utente di recuperare la password
  + Logout – consente all’utente di uscire dal sito
  + Login – consente all’utente di loggarsi al sito
* Gestione Amministratore
  + AggiungiLibro – consente all’amministratore di aggiungere un libro alla libreria
  + ModificaLibro - consente all’amministratore di modificare un libro nella libreria
  + RimuoviLibro - consente all’amministratore di rimuovere un libro dalla libreria
  + BanUser - consente all’amministratore di bannare un utente dal sito
  + VisualizzaListaLibri - consente all’amministratore di visualizzare la libreria
  + VisualizzaListaUtenti - consente all’amministratore di visualizzare la lista degli utenti
  + VisualizzaListaSegnalazioni - consente all’amministratore di visualizzare la lista segnalazioni
  + VisualizzaSegnalazione - consente all’amministratore di visualizzare una singola segnalazione
* Gestione Moderatore
  + ModificaFotoCommunity – consente al moderatore di modificare la foto di una sua community
  + ModificaCommunity – consente al moderatore di modificare I dati di una sua community
  + EliminaCommunity – consente al moderatore di eliminare una sua community
  + VisualizzaUtenti – consente al moderatore di visualizzare la lista utenti di una sua community
* Gestione HomePage
  + Ricerca – consente all’utente di ricercare all’interno del sito
* Gestione Post
  + CaricaPost – consente all’utente di caricare un post in una community
  + ModificaPost – consente all’utente di modificare un proprio post in una community
  + InserisciCommento – consente all’utente di inserire un commento in un post
  + NuovaSegnalazionePost – consente all’utente di segnalare un post
  + Like – consente all’utente di aggiungere una valutazione positiva a un post
  + RemoveLike – consente all’utente di rimuovere la propria valutazione positiva a un post
  + VisualizzaPost – consente all’utente di visualizzare un post
  + RimuoviPost – consente al moderatore o all’utente di rimuovere un post
* Gestione Profilo
  + ModificaPassword – consente all’utente di modificare la password del proprio account
  + ModificaProfilo – consente all’utente di modificare I dati del proprio profilo
  + ModificaImmagine – consente all’utente di modificare l’immagine del proprio profilo
  + EliminaAccount – consente all’utente di eliminare il proprio account
  + VisualizzaProfilo – consente all’utente di visualizzare il proprio profilo o quello di un altro utente
* Gestione Libro
  + VisualizzaLibro – consente all’utente di visualizzare I dati di un libro
  + VisualizzaCommunity – consente all’utente di visualizzare una community
  + VisualizzaLibri – consente all’utente di visualizzare la lista libri
  + FollowCommunity – consente all’utente di seguire una community e tutti I post relativi ad essa
  + UnfollowCommunity – consente all’utente di levare il follow ad una community e tutti I post relativi ad essa
  + AllcommunityByLibro – consente all’utente di visualizzare tutte le community di un determinato libro
  + creaCommunity – consente all’utente di creare una community