



UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea in Informatica
Progetto Ingegneria del Software

SYSTEM DESIGN DOCUMENT

VERSIONE 1.2



BookPad

Partecipanti

Nome	Matricola
Marica D'Alfonso	0512106258
Marianna Farina	0512109126

Sommario

Sommario	2
1. Introduzione.....	3
1.1 Scopo del sistema	3
1.2 Obiettivi del sistema	3
1.3 Definizione, acronimi e abbreviazioni.....	3
1.4 Riferimenti	3
1.5 Panoramica	4
2. Architettura Software Attuale	4
3. Architettura Software Proposta.....	4
3.1 Panoramica	4
3.2 Decomposizione in sottosistemi	4
3.3 Mappatura Hardware/Software	6
3.4 Gestione dei dati persistente.....	7
3.5 Controllo degli accessi e sicurezza	9
3.6 Controllo globale del software.....	10
4. Servizi dei sottosistemi	10
4.1 Gestione Capitoli.....	10
4.2 Gestione Commenti	10
4.3 Gestione Generi	10
4.4 Gestione Utenti.....	10
4.5 Gestione Storie.....	11
4.6 Gestione Tag	11

1. Introduzione

1.1 Scopo del sistema

“BookPad” è una piattaforma dedicata alla scrittura ed alla lettura. Il suo scopo principale è quello di permettere alle persone che hanno queste passioni di cimentarsi nella stesura di storie e/o di leggere centinaia di storie pubblicate da altri utenti della piattaforma. Inoltre, permette l’interazione diretta tra scrittori e lettori con la possibilità di commentare le storie ed inserire likes. Essere iscritti a BookPad non significa solo dover scrivere o leggere una storia, ma anche avere la possibilità di conoscere altri utenti con le stesse passioni.

1.2 Obiettivi del sistema

Il sistema “BookPad” è stato progettato considerando i seguenti obiettivi di design:

1.2.1 Criteri di affidabilità

- **Sicurezza:** Il sistema deve garantire la protezione dei dati sensibili scambiati tra utente e sistema, in particolare la password deve essere criptata quando salvata all’interno del database
- **Robustezza:** Il sistema deve essere in grado di gestire correttamente l’immissione di eventuali input errati, in particolare ogni volta che l’utente compila degli input, l’input deve essere controllato in modo da non permettere la sottomissione di input non ammesso.

1.2.2 Criteri di Manutenzione

- Il sistema deve adottare il modello MVC in modo da favorire una progettazione del software più modulare, rendendo più facile l'**estensibilità** e la **modificabilità** del software.
- **Adattabilità:** l'adattabilità si riferisce alla capacità del software di adattarsi a diverse situazioni e contesti di utilizzo. Il sistema deve quindi disporre della modalità responsive. In questo modo l’applicazione si può adattare a diversi dispositivi e dimensioni di schermo, garantendo un'esperienza d'uso ottimale su ogni dispositivo.

1.2.3 Criteri dell’utente finale

- **Usabilità:** il sistema deve prevedere una dark mode che può aiutare a migliorare l'accessibilità dell'applicazione, rendendola più facile da utilizzare per gli utenti con problemi di vista o per chi lavora in ambienti con poca illuminazione. In questo modo si può migliorare l'esperienza d'uso dell'applicazione, aumentando la soddisfazione degli utenti.

1.3 Definizione, acronimi e abbreviazioni

RAD sta per Requirement Analysis Document.

SDD sta per System Design Document.

MVC sta per Model View Control.

JSP sta per Java Servlet Page.

1.4 Riferimenti

Riferimento al Requirement Analysis Document di BookPad.

1.5 Panoramica

Il seguente System Design Document (SDD) mostra i dettagli tecnici del design del sistema “BookPad”. Altri dettagli riguardanti le funzionalità e le caratteristiche del sistema possono essere trovati nel Requirement Analysis Document (RAD), mentre una descrizione più generale del sistema può essere trovata nel Problem Statement.

- ❖ Nel capitolo 2 viene mostrata l’architettura del sistema corrente;
- ❖ Nel capitolo 3 viene mostrata l’architettura del sistema proposto, in particolare:
 - Decomposizione del sistema in sottosistemi di funzionalità e operazioni correlate;
 - Mapping Hardware/Software;
 - Gestione dei dati persistenti;
 - Controllo degli accessi e della sicurezza;
 - Controllo del flusso globale;
- ❖ Nel capitolo 4 vengono mostrati i servizi forniti da ogni sottosistema.

2. Architettura Software Attuale

L’architettura software attuale non è esistente.

3. Architettura Software Proposta

3.1 Panoramica

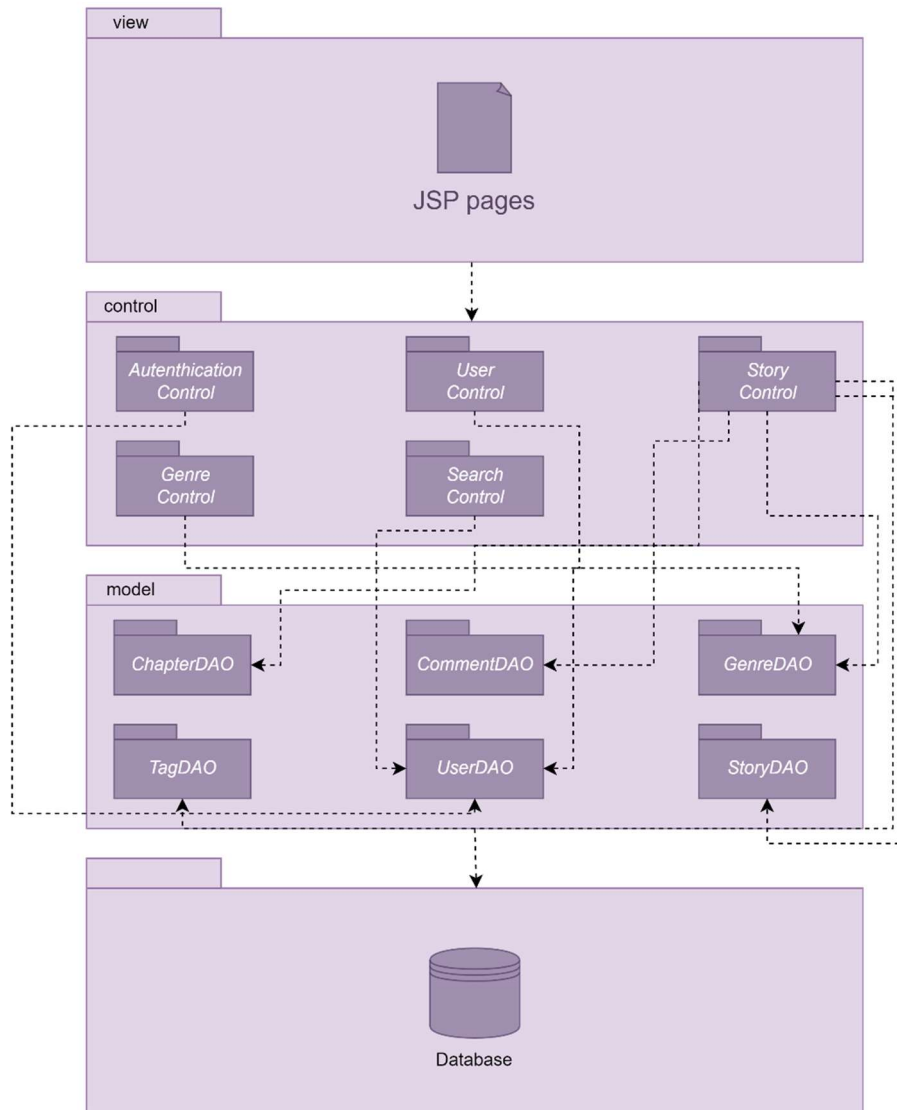
Di seguito verrà mostrata la suddivisione del sistema in sottosistemi con i relativi compiti che ogni sottosistema dovrà offrire e il mapping hardware/software del sistema.

Inoltre, vengono descritti i dati persistenti con il relativo schema relazionale, il controllo degli accessi con la descrizione degli utenti del nostro sistema, ed il controllo generale del software.

3.2 Decomposizione in sottosistemi

Il sistema adotta l’architettura MVC (Model View Control). I sottosistemi principali previsti dall’architettura MVC sono:

- **Model:** rappresenta il sistema di gestione dei dati. Il modello comunica con il database o altre fonti dati per recuperare, aggiornare o eliminare le informazioni.
- **View:** rappresenta l’interfaccia utente dell’applicazione; La view visualizza i dati dell'applicazione in modo che l'utente possa interagire con essi.
- **Controller:** gestisce le richieste dell'utente e coordina l'interazione tra model e la view. Il controller riceve i dati dall'input dell'utente, elabora le richieste e aggiorna il model e la view in base alle azioni dell'utente.

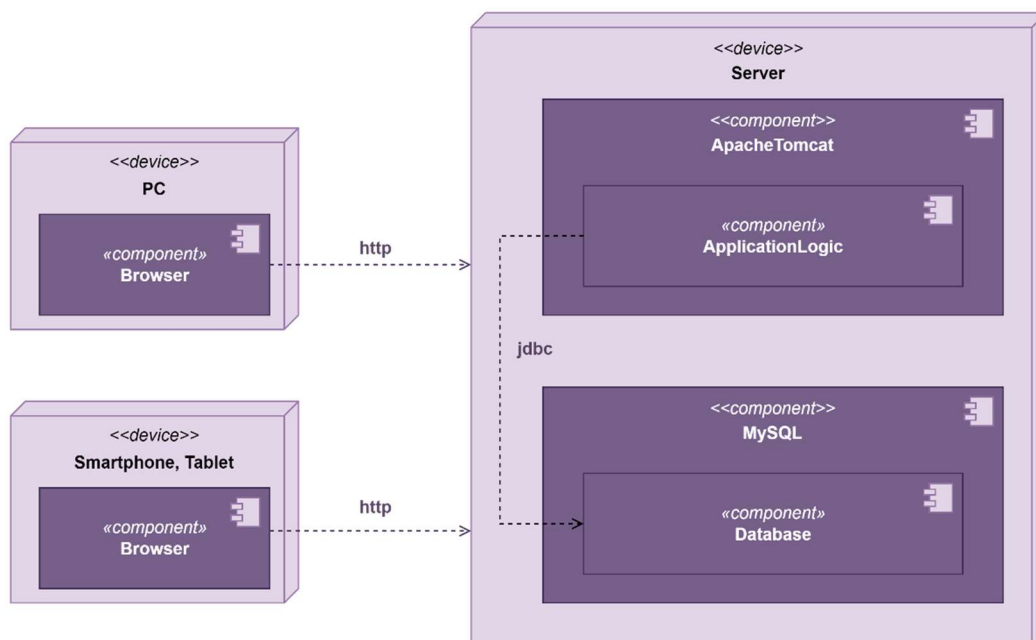


ChapterDAO	Permette di: <ul style="list-style-type: none"> ▪ Aggiungere i capitoli ad una storia che si sta salvando ▪ Ottenere tutti i capitoli appartenenti ad una storia ▪ Ottenere il capitolo che si vuole visualizzare
CommentDAO	Permette di: <ul style="list-style-type: none"> ▪ Aggiungere i commenti ad una storia ▪ Ottenere tutti i commenti appartenenti ad una storia
GenreDAO	Permette di: <ul style="list-style-type: none"> ▪ Ottenere tutti i generi presenti sulla piattaforma ▪ Ottenere le informazioni di un singolo genere
UserDAO	Permette di: <ul style="list-style-type: none"> ▪ Far loggare un utente alla piattaforma ▪ Far registrare un utente alla piattaforma ▪ Ottenere le informazioni di un utente ▪ Ottenere i followers ed i followings di un utente

	<ul style="list-style-type: none"> ▪ Ricercare utenti tramite username ▪ Seguire/Smettere di seguire un utente ▪ Modificare l'e-mail, la password, la biografia e l'avatar di un utente
StoryDAO	Permette di: <ul style="list-style-type: none"> ▪ Pubblicare una storia ▪ Ricercare storie tramite tag, titolo o genere ▪ Ottenere le informazioni di una storia ▪ Aggiungere/Rimuovere il like ad una storia ▪ Salvare/Eliminare una storia ▪ Ottenere tutte le storie salvate ▪ Ottenere tutte le storie scritte
TagDAO	Permette di: <ul style="list-style-type: none"> ▪ Ottenere tutti i tag di una storia ▪ Aggiungere tag ad una storia che si sta salvando

3.3 Mappatura Hardware/Software

Il sistema utilizza un'architettura Client/Server. In particolare, Apache Tomcat 9 è il Web Server, la logica del sistema è costituita da Java Servlet, l'interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page), il Web Browser utilizzato dall'utente rappresenta il Client, un database Server MySQL ospiterà i dati persistenti.



3.4 Gestione dei dati persistente

3.4.1 Descrizione delle entità persistenti

User

- id: int(PK)
- username: VARCHAR(20)
- email: VARCHAR(256)
- password: VARCHAR(256)
- about: VARCHAR(500)
- avatar: LONGBLOB

La tabella User contiene le informazioni riguardanti gli utenti iscritti alla piattaforma.

Storie

- id: int(PK)
- title: VARCHAR(50)
- plot: MEDIUMTEXT
- cover: LONGBLOB
- n_like: int
- n_comments: int
- n_savings: int
- dt: DATETIME
- ID_genre: int(FK)
- ID_user: int(FK)

La tabella Storie contiene le informazioni circa le storie pubblicate sulla piattaforma.

Genre

- id: int(PK)
- name: VARCHAR(50)
- description: VARCHAR(300)

La tabella Genre contiene le informazioni circa i generi disponibili sulla piattaforma.

Chapter

- n_chap: int(PK)
- ID_story: int(PK, FK)
- title: VARCHAR(50)
- content: MEDIUMTEXT

La tabella Chapter contiene le informazioni circa i capitoli che compongono le storie della piattaforma.

Comment

- id: int(PK)
- content: TEXT
- data_cm: DATE
- ID_user: int(FK)
- ID_story: int(FK)

La tabella Comment contiene le informazioni circa i commenti appartenenti alle storie della piattaforma.

Follow

- ID_who_i_follow: int(PK,FK)
- ID_followed_by: int(PK,FK)

La tabella Follow indica le associazioni tra i diversi utenti che si seguono.

Has_like

- ID_user: int(PK,FK)
- ID_story: int(PK,FK)

La tabella Has_like indica le associazioni tra i diversi utenti e le storie a cui hanno messo like.

Library

- ID_user: int(PK,FK)
- ID_story: int(PK,FK)

La tabella Library indica le associazioni tra i diversi utenti e le storie che hanno salvato.

Tag

- Text_tag: VARCHAR(500)(PK)

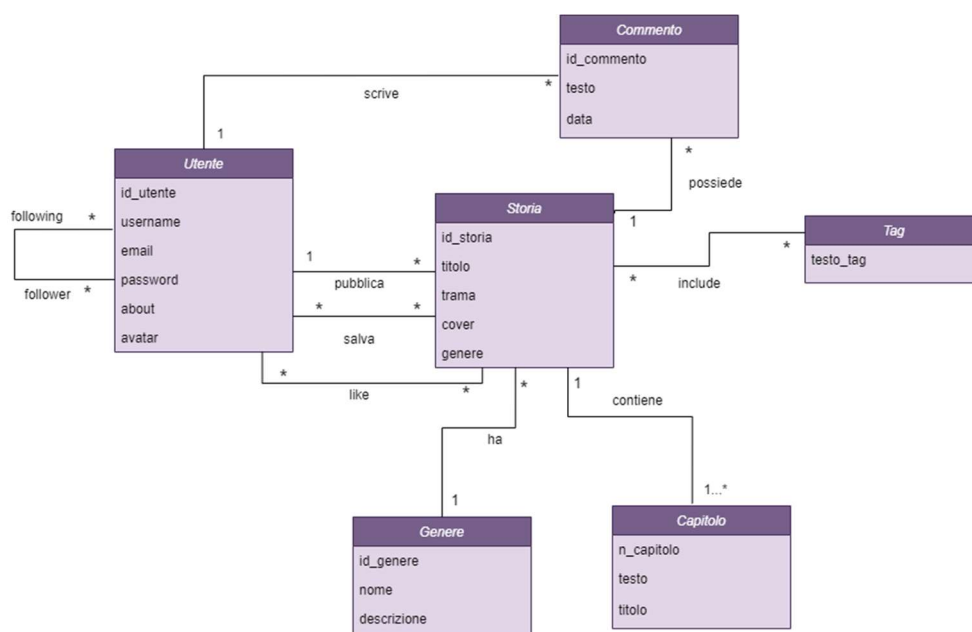
La tabella Tag contiene i tags delle piattaforma

Has_tag

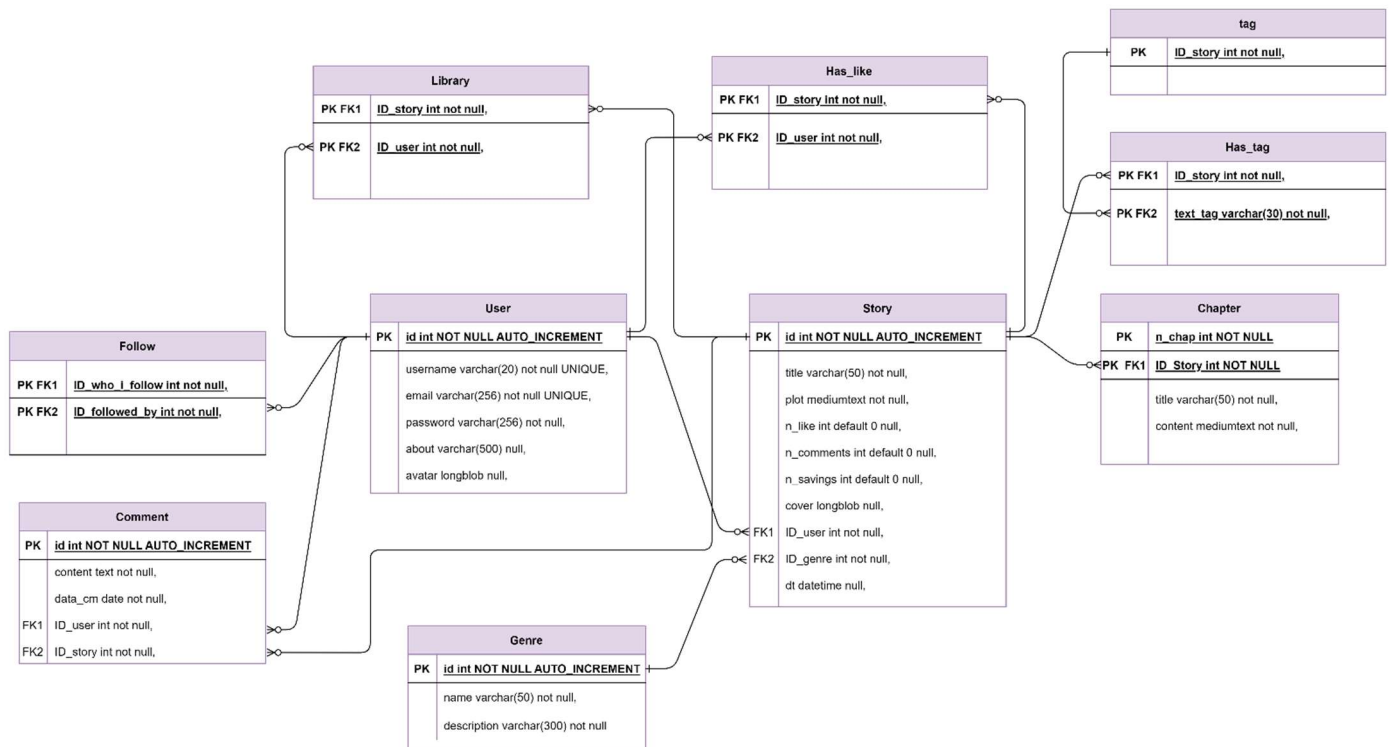
- ID_story: int(PK,FK)
- Text_tag: VARCHAR(500)(PK)

La tabella Has_tag indica le associazioni tra le diverse storie e i tag che possiedono

3.4.2 Diagramma dei dati persistenti



3.4.3 Schema Logico



3.5 Controllo degli accessi e sicurezza

Il controllo degli accessi è garantito tramite l'utilizzo di e-mail e password per gli utenti del sistema che hanno possibilità di creare o modificare gli oggetti che modellano entità di dominio, così da prevenire accessi non autorizzati ad informazioni sensibili.

Il sistema non fornirà un metodo di recupero della password.

Si ricorrerà all'utilizzo della sessione per tenere traccia dell'utente loggato.

Il salvataggio delle password nel database sarà cifrato.

Le operazioni che gli utenti dell'applicazione web possono effettuare sugli oggetti sono riportate nella tabella che segue:

	User	Story	Area Personale
Utente registrato	<ul style="list-style-type: none"> - Visualizzare - Ricercare - Seguire - Smettere di seguire 	<ul style="list-style-type: none"> - Ricercare - Pubblicare - Salvare - Mettere like - Commentare 	<ul style="list-style-type: none"> - Login - Modificare - Logout
Utente Guest	<ul style="list-style-type: none"> - Visualizzare - Ricercare 	<ul style="list-style-type: none"> - Ricercare 	<ul style="list-style-type: none"> - Registrare

3.6 Controllo globale del software

BookPad è un'applicazione web che si basa su un Web Server per ricevere e gestire le richieste effettuate dagli utenti. Il server si occupa di indirizzare le richieste ai Control (Java Servlet) pertinenti, i quali si occupano di elaborare le richieste e, se necessario, di interagire con il Model. Successivamente, il Web Server si incarica di aggiornare le View, ovvero le pagine che verranno mostrate all'utente. Questo avviene attraverso la creazione di codice HTML a partire dalle pagine JSP.

4. Servizi dei sottosistemi

È stato deciso di dividere il sistema in sottosistemi, ognuno avente un insieme di funzionalità.

4.1 Gestione Capitoli

Elenco capitoli	Consente di ottenere l'elenco dei capitoli
Ottieni capitolo	Consente di ottenere il contenuto di un capitolo
Aggiungi capitolo	Consente di aggiungere un capitolo ad una storia

4.2 Gestione Commenti

Aggiungi commento	Consente di aggiungere commenti ad una storia
Elenco commenti	Consente di ottenere l'elenco dei commenti

4.3 Gestione Generi

Elenco generi	Consente di ottenere l'elenco dei generi della piattaforma
Ottieni genere	Consente di ottenere le informazioni di un genere

4.4 Gestione Utenti

Login	Consente all'utente di accedere al sistema per poter svolgere le operazioni a lui consentite
Logout	Consente all'utente di uscire dal sistema
Registrazione	Consente all'utente di registrarsi nel sistema
Ottieni utente	Consente di ottenere le informazioni di un utente
Elenco followers	Consente di ottenere l'elenco dei followers di un utente
Elenco followings	Consente di ottenere l'elenco dei followings di un utente
Modifica biografia	Consente all'utente di modificare la propria biografia
Modifica avatar	Consente all'utente di modificare il proprio avatar
Modifica e-mail	Consente all'utente di modificare la propria e-mail inserita nel sistema

Modifica password	Consente all'utente di modificare la propria password inserita nel sistema
Follow	Consente all'utente di salvare una o più storie di altri utenti
Unfollow	Consente all'utente di eliminare una o più storie in precedenza salvate
Ricerca utenti	Consente di cercare utenti, attraverso l'username, iscritti alla piattaforma

4.5 Gestione Storie

Pubblica storia	Consente all'utente di inserire all'interno del sistema la propria storia con le informazioni relative ad essa e i capitoli
Salva storia	Consente all'utente di salvare una o più storie di altri utenti
Eliminare storia salvata	Consente all'utente di eliminare una o più storie in precedenza salvate
Ottieni storia	Consente di ottenere le informazioni circa una storia
Mettere like	Consente all'utente di mettere like alla storia
Rimuovere like	Consente all'utente di rimuovere like ad una storia a cui si era messi like in precedenza
Ricerca storie	Consente di cercare storie, attraverso tag o titolo, pubblicate sulla piattaforma
Elenco storie followings	Consente di ottenere le storie pubblicate dai propri followings
Elenco storie salvate	Consente di ottenere le storie salvate da un utente
Elenco storie pubblicate	Consente di ottenere le storie pubblicate da un utente

4.6 Gestione Tag

Aggiungi tag	Consente di aggiungere i tag ad una storia
Elenco tags	Consente di ottenere l'elenco dei tags