

1. Descrizione del Sistema

Analisi della situazione reale.

L'applicazione deve simulare un vecchio registro cartaceo dei prestiti librari di un istituto scolastico, il quale deve essere dismesso e trasformato in un registro interamente digitalizzato ed intitolato "**BiblioTech**".

L'applicazione dovrà gestire il ciclo di vita dei vari prestiti forniti agli utenti, andando a monitorare in tempo reale le giacenze di magazzino dei medesimi e differenziando le operazioni in base al ruolo dell'utente.

Essa inoltre deve occuparsi della gestione del patrimonio librario nel quale vi sono contenuti diversi titoli di libri che possono essere presenti in un quantitativo maggiore o uguale ad uno nel registro della scuola.(si potrebbe pensare anche di inserire un titolo che non ha copie presenti nel database fisico e che quindi risulterebbe presente nella rubrica digitale,ma che alla fine non può essere reperito dagli stessi utenti).

Per ogni Titolo in sistema digitalizzato dovrà conoscere il numero di copie TOTALI che sono possedute dal sistema e il numero di copie che risulterebbero attualmente disponibili per il prestito, quindi andando ad eliminare nel conteggio delle copie totali quelle che sono già state emanate in prestito.

L'applicazione deve gestire l'accesso per le varie utenze presenti all'interno di essa.

nel sistema risulteranno soltanto due tipi di bacini di utenza che sono:

- **Studenti** : coloro che possono visualizzare il catalogo con tutti i libri presenti del database della libreria e hanno la possibilità tramite un click di poter effettuare un prestito di un libro a loro piacere soltanto nel caso in cui vi sia almeno una copia di esso disponibile.
Inoltre devono poter visualizzare lo storico dei libri da loro presi in prestito.
- **Bibliotecari** : coloro che hanno una visione globale dell'intero sistema, avendo la possibilità di visualizzare l'elenco dei libri in prestito e in caso registrarne la loro restituzione.

Tramite l'operazione di prestito da parte di uno studente si deve tener traccia delle copie disponibili attraverso il decurtamento del contatore che registra il valore di esse, invece durante la restituzione, e quindi l'operazione che può

essere svolta dal bibliotecario il contatore delle copie disponibili deve essere incrementato.

Si ipotizzi inoltre che ogni studente può effettuare un massimo di due prestiti per volta e una volta preso in prestito un determinato titolo avrà un tempo determinato per restituirlo, periodo che dura circa un mese, in caso dovesse non rispettare la data di scadenza lo studente sarà soggetto ad una multa amministrativa che avrà un importo che varierà in base al tempo passato dalla scadenza.

Ad esempio :

- 2 € se si supera la data di scadenza di un giorno
- 5 € se si supera di una settimana
- 10 € se si supera di un mese

Pertanto sarebbe opportuni anche inserire una variabile nell'utente che prevederà il totale costo monetario che lui ha a disposizione nel suo portafoglio, che quindi verrebbe utilizzato nel caso in cui non andrebbe a rispettare la data di scadenza del prestito.

Per simulare il tutto però ho deciso di cambiare il tempo limite per la restituzione di un libro perchè simulare l'attesa di un mese sarebbe risultato difficoltoso e quindi ho modificato il tempo ad:

- 2 € se non si restituisce il libro entro 5 minuti
- 5 € se non si restituisce entro 10 minuti
- 10 € se si supera il limite di 10 minuti

Si ipotizzi inoltre una caso nella quale uno studente andrebbe a richiedere un determinato titolo che attualmente risulterebbe non disponibile per mancanza di copie disponibili e tramite una variabile contenente l'email dello studente interessato esso verrebbe avvisato tramite questa modalità su una futura disponibilità di una copia del titolo scelto inizialmente, andando quindi a registrare delle richieste da parte degli studenti.

Componenti Principali

- Patrimonio Librario: Ogni libro ha un titolo, autore, ISBN univoco, copie totali (fisiche possedute) e copie disponibili (totali - prestiti attivi). Le copie disponibili vengono aggiornate in tempo reale su prestito/restituzione.

→ Utenti e Ruoli:

- ◆ Studenti: Visualizzano catalogo e dettagli libri. Possono prendere in prestito (max 2 attivi) se disponibili, visualizzare prestiti attivi. Possono richiedere libri non disponibili (notifica simulata).
- ◆ Bibliotecari: Visualizzano tutti prestiti attivi, registrano restituzioni (con calcolo multe automatico).

→ Movimentazione:

- ◆ Prestito: Decrementa copie disponibili, crea record in prestiti con data inizio e scadenza (1 mese).
- ◆ Restituzione: Chiude prestito, incrementa copie disponibili, calcola/detrae multe dal portafoglio.
- ◆ Richieste: Studenti richiedono libri non disponibili; notificati (simulato) quando disponibili.

→ Sicurezza: Autenticazione con login/logout sicuri; controlli ruolo per accesso; protezione SQL injection e hashing password.

→ Ulteriori Specifiche/Customizzazioni:

- ◆ Max 2 prestiti attivi per studente (bloccato se superato).
- ◆ Scadenza automatica (1 mese); multe progressive (5€/giorno, 10€/settimana, 20€/mese) detratte dal portafoglio (se < multa, prestito bloccato).
- ◆ Portafoglio utente per multe (iniziale 50€ per studenti, 0€ per bibliotecari).
- ◆ Nessun 2FA o API (focus su base); interfaccia responsive con Bootstrap.
- ◆ Gestione errori: Messaggi user-friendly; logging simulato in console per debug.

Flusso Operativo

1. Utente si registra/logga.
2. Studente: Naviga catalogo, prende in prestito se disponibile (max 2), vede prestiti attivi.
3. Bibliotecario: Vede prestiti attivi, restituisce (con multe).
4. Sistema aggiorna DB in tempo reale.

Architettura

- Frontend: HTML/CSS/Bootstrap per UI.
- Backend: PHP procedurale per logica business.
- Database: MySQL/MariaDB con tabelle normalizzate.
- Sicurezza: Sessioni PHP; controlli ruolo; HTTPS consigliato.

2. Diagramma E-R (Entity-Relationship)

Lo Schema ER (o Diagramma Entità-Relazione) è un modello grafico usato per progettare e rappresentare la struttura di un database relazionale. Serve a visualizzare le entità (oggetti del mondo reale), i loro attributi (caratteristiche) e le relazioni tra di esse, prima di implementare il database in SQL. Lo scopo principale di questo schema è quello di aiutare a capire come organizzare i dati, identificare chiavi primarie/esterne e cardinalità (quanti elementi possono essere collegati). È essenziale per evitare ridondanze e garantire integrità.

IL DIAGRAMMA E/R è PRESENTE NELLA SEZIONE [bibliotech/docs](#)

Il nome del file è [BiblioTechE_R.drawio \(1\).pdf](#)

3. Diagramma delle Classi (UML)

L'UML (Unified Modeling Language) è un linguaggio di modellazione standardizzato per descrivere sistemi software, inclusi aspetti strutturali e comportamentali. È usato in ingegneria del software per progettare architetture, classi e interazioni, prima di scrivere codice.

Fornisce diagrammi visivi per rappresentare la logica del sistema, facilitando comunicazione tra sviluppatori e stakeholders. Non è legato a un linguaggio specifico (può essere usato per PHP, Java, ecc.).

IL DIAGRAMMA UML è PRESENTE NELLA SEZIONE [bibliotech/docs](#)

Il nome del file è [BiblioTech UML.drawio.pdf](#)

4. Specifiche di Sessione e Sicurezza

Al momento del login riuscito, il sistema salva specifici dati nella superglobale `$_SESSION` di PHP.

Questi dati sono essenziali per mantenere lo stato dell'utente durante la navigazione, personalizzare l'interfaccia e applicare controlli di sicurezza.

La tabella seguente elenca ciascun dato, con descrizione, tipo, scopo e considerazioni di sicurezza.

I dati vengono impostati solo dopo la verifica positiva della password (tramite `password_verify()`) e l'inserimento di una nuova riga nella tabella sessioni del database, che traccia le sessioni attive.

NOME	DESCRIZIONE	TIPO	SCOPO
idU	ID univoco dell'utente	int	Identifica l'utente in query successive
nome	Nome dell'utente	string	Personalizza l'interfaccia utente
ruolo	Ruolo dell'utente	string	Determina i permessi di accesso
logged_in	Flag booleano che indica lo stato di autenticazione	boolean	Verifica rapida se l'utente è autenticato
idSess	ID della sessione attiva	int	Traccia la sessione per audit

Il sistema BiblioTech implementa un modello di controllo di accesso basato su ruoli per impedire a utenti con ruolo "studente" di accedere a risorse riservate ai "bibliotecari".

Questo è realizzato attraverso controlli lato server in PHP, utilizzando i dati di sessione salvati al login.

Logica di Funzionamento:

Ogni pagina riservata inizia con un controllo che verifica lo stato di autenticazione e il ruolo dell'utente. Se l'utente non è loggato o non ha il ruolo corretto, viene reindirizzato a una pagina sicura (es. `index.php` o `login.php`).

Questo previene l'esecuzione di codice riservato, come query DB per gestione prestiti o visualizzazione dati amministrativi.

