

DocUfficiale

1) RACCOLTA/ANALISI DEI REQUISITI

A) TESTO COMPLETO SULLA SPECIFICA DEI DATI

ENTITA'

BIBLIOTECA

- Di un nome(univoco), un indirizzo, un'email, un sito web delle coordinate (latitudine/longitudine), uno o più recapiti telefonici, un campo di testo relativo alle note storiche.
- Ogni biblioteca può disporre di un numero limitato di posti lettura.
- Ogni biblioteca dispone di un numero limitato di posti lettura.
- Una biblioteca UNIBO può avere più amministratori.

POSTO LETTURA

- Ogni posto lettura dispone di un numero progressivo (univoco, ma solo all'interno di una biblioteca), dell'indicazione se dotato di presa di corrente o meno (campo booleano) e dell'indicazione se dotato di presa di rete Ethernet o meno (campo booleano).

LIBRI

- La biblioteca mette a disposizione del pubblico l'accesso ad i proprio libri.
- Ogni libro dispone di un codice (univoco a livello UNIBO), un titolo, una lista degli autori, un anno di pubblicazione, un nome dell'edizione, un genere.
- I libri possono appartenere a due categorie (e solo a quelli): libri cartacei o ebook.

LIBRI CARTACEI

- Nel primo caso, si vogliono memorizzare anche stato di conservazione, numero di pagine, numero di scaffale, e stato del prestito.
- Lo stato di conservazione può assumere solo quattro valori: ottimo, buono, non buono, scadente.
- Lo stato del prestito può essere: disponibile, prenotato e consegnato.

EBOOK

- Nel caso degli e-book, si vogliono memorizzare anche la dimensione, il numero di accessi alla scheda e il PDF del documento.

UTENTE

- Ogni utente dispone di e-mail, password, nome, cognome, data di nascita, luogo di nascita, recapito telefonico.
- Gli utenti possono appartenere a tre categorie: amministratori di biblioteca, volontari e utilizzatori.

AMMINISTRATORE

- Gli amministratori (dipendente UNIBO) dispongono anche del campo qualifica (testo di massimo 10 caratteri).
- Ogni amministratore è responsabile di una sola biblioteca UNIBO.
- Gli amministratori possono inviare segnalazioni per comportamenti non corretti da parte di utilizzatori.

VOLONTARI

- Gli utenti volontari hanno un campo mezzo di trasporto che può essere: piedi, bici, auto.
- Gli utenti volontari si fanno carico di consegnare i libri prenotati agli utilizzatori.

EVENTO CONSEGNA

- Ogni evento di consegna è inserito da un utente volontario, fa riferimento ad una prenotazione di testo cartaceo.
- Può essere di tipo “Restituzione” o “Affidamento” e dispone di un campo data e note (massimo 200 caratteri).

UTILIZZATORI

- Gli utenti utilizzatori dispongono di un campo aggiuntivo relativo alla data di creazione dell’account, un campo professione ed un campo relativo allo stato dell’account.
- Gli utilizzatori possono prenotare un posto lettura presso una biblioteca UNIBO.
- Gli utilizzatori possono accedere liberamente agli ebook disponibili: tuttavia il sistema tiene traccia dello storico degli accessi agli ebook (o meglio alle loro schede) effettuati da ciascun utente
- In maniera simile, gli utilizzatori possono prenotare un libro cartaceo, a patto che il testo sia nello stato “Disponibile”, e che lo stato di conservazione non sia pari a “Scadente”.
- Nel caso in cui un utilizzatore riceva cumulativamente più di 3 segnalazioni (anche da amministratori di biblioteche diverse), lo stato dell’account viene settato a “Sospeso” (impedendo qualsiasi accesso alla piattaforma da parte dell’utente selezionato).

PRENOTAZIONE

- Ogni prenotazione dispone di un campo data, ora inizio ed ora fine.
- La prenotazione di un posto lettura è possibile solo a condizione che la biblioteca abbia effettivamente dei posti lettura disponibili per la data /orario richiesto.
- La prenotazione dispone di un codice (univoco), una data di avvio e una data di fine (automaticamente settata a +15 giorni a partire dalla data di consegna).

MESSAGGIO

- Ogni messaggio è inserito da un amministratore ed è destinato ad un utente utilizzatore, e dispone di un titolo (ex: “Libro non disponibile”), un campo testo ed una data.

SEGNALAZIONE

- Ogni segnalazione dispone di una data ed eventuale nota di testo ed è inserita da un amministratore e diretta verso un utilizzatore.

B) LISTA DELLE OPERAZIONI

UTILIZZATORE

- **PrenotaPosto** = Prenota un posto lettura all'interno di una biblioteca
- **PrenotaLibro** = Prenota un libro cartaceo
- **PrenotazioniLibroUtilizzatore** = Storico delle prenotazioni dei libri
- **PostoPrenotatoUtilizzatore** = Storico delle prenotazioni posto
- **VisualizzaEbook** = Visualizza ebook dal codice
- **EventiConsegnaUtilizzatori** = Storico degli eventi consegna
- **PrenotazioniLibriAttivi** = Prenotazioni libri in corso
- **PostiPrenotatiAttivi** = Prenotazioni posto in corso
- **EventiConsegnaAttivi** = Consegne ancora da ricevere
- **SegnalazioniRicevute** = Visualizza le segnalazioni ricevute
- **MessaggiRicevuti** = Visualizza i messaggi ricevuti
- **EliminaPrenotazionePosto** = Elimina la prenotazione di un posto
- **EliminaPrenotazioneLibro** = Elimina la prenotazione di un libro
- **EbookDisponibili** = Visualizza ebook presenti nel sistema
- **LibriBiblioteca** = Catalogo libri di una biblioteca
- **EbookDalNome** = Ricerca informazioni tramite titolo di un ebook
- **LibroDalNome** = Ricerca informazioni tramite titolo di un cartaceo
- **CambiaConsegnaUtilizzatore** = Cambia la data di una consegna
- **LibriDisponibili** = Mostra tutti i libri prenotabili nel sistema
- **RegistrazioneUtente** = Registra un nuovo utilizzatore

VOLONTARIO

- **PrenotazioniTotaliVolontario** = Storico delle consegne effettuate
- **NuovoEvento** = Nuovo evento di consegna

- **AggiornamentoEventoConsegna** = Aggiorna un evento di consegna
- **PrenotazioniVolAttive** = Consegne ancora da portare a termine
- **ConsegneDisponibiliAffidamento** = Mostra le consegne di tipo “**Affidamento**” che può fare
- **ConsegneDisponibiliRestituzione** = Mostra le consegne di tipo “**Restituzione**” che può fare
- **RegistrazioniVolontario** = Regista un nuovo volontario

AMMINISTRATORE

- **CatalogaLibro** = Aggiunge nuovo libro in una biblioteca
- **EliminaLibro** = Elimina un libro dalla biblioteca
- **AggiornamentoLibro** = Aggiorna i dati di un libro all'interno di una biblioteca
- **PrenotazioniBiblioteca** = Storico delle prenotazioni
- **MandaMessaggio** = Invia messaggio all'utente utilizzatore
- **InviaSegnalazione** = Invia segnalazione all'utente utilizzatore
- **RimuoviSegnalazione** = Rimuove una segnalazione da un utente
- **PrenotazioniLibriAmministratoreAttive** = Visualizza prenotazioni dei libri attivi
- **PostiBibliotecaStorico** = Visualizza i posti prenotati della tua biblioteca
- **PostiBibliotecaAttivi** = Visualizza prenotazioni dei libri attivi

INFORMAZIONI

- **VisualizzaBiblioteche** = Visualizza tutte le informazioni relative alla biblioteca
- **NumeroPostiLetturaBiblioteca** = Visualizza tutti i posti lettura delle biblioteche
- **TuttiILibri** = Visualizza tutti i libri all'interno della biblioteca

STATISTICHE

- **ClassificaVolontari** = Classifica volontari in base al numero di consegne effettuate
- **ClassificaCartacei** = Classifica dei libri cartacei in base al numero delle prenotazioni
- **ClassificaEbook** = Classifica degli ebook sulla base del numero di letture di ciascuno
- **BibliotecheMenoUtilizzate** = Classifica delle biblioteche che hanno meno posti usati rispetto alla loro capacità.
- **VolontarioDelMese** = Volontario che ha effettuato più consegne in quel determinato mese
- **LibroDelMese** = Libro che ha avuto più prenotazioni in un determinato mese

C) TAVOLA MEDIA DEI VOLUMI

CONCETTO	TIPO	VOLUME
Biblioteca	E	5
Utente	E	2555
Amministratore	E	5
Volontario	E	50
Utilizzatore	E	2500
Recapito telefonico	E	10
Galleria immagini	E	5
Posto lettura	E	100
Libro	E	25000
Cartaceo	E	10000
Ebook	E	15000
Prenotazione libro	E	5000
Segnalazione	E	100
Messaggio	E	200
Prenotazione posto	R	4800
Evento consegna	R	4000
Lettura	R	5000

La tabella dei volumi è uno strumento che serve per calcolare il costo delle operazioni e per l'analisi della ridondanza. Si nota che è scelto di adottare cinque biblioteche e che tutti i dati sono stati scelti cercando di mantenere un'omogeneità tra di loro.

ASSUNZIONI

Per completare la tabella dei volumi è necessario conoscere una stima delle mole di dati di ogni tabella. La scrittura delle assunzioni serve per esplicitare tutte le scelte per quanto riguarda il valore di ogni stima. Di seguito l'elenco con tutte le assunzioni:

- **Assunzione 1:** un amministratore per ogni biblioteca
- **Assunzione 2:** 20 volontari per ogni biblioteca
- **Assunzione 3:** 500 utilizzatori per ogni biblioteca
- **Assunzione 4:** 2 recapiti telefonici per ogni biblioteca
- **Assunzione 5:** 20 posti lettura (numero basso dovuto alla necessità di distanziamento) per ogni biblioteca
- **Assunzione 6:** 2000 libri cartacei ogni biblioteca
- **Assunzione 7:** 3000 ebook per ogni biblioteca
- **Assunzione 8:** 1000 prenotazioni di libri cartacei al mese per ogni biblioteca
- **Assunzione 9:** i 20 posti di ognuna delle 5 biblioteche vengono prenotati 2 volte al giorno (mattina e pomeriggio) per 24 giorni al mese (tutta la settimana tranne la domenica in cui la biblioteca è chiusa)
- **Assunzione 10:** 20 segnalazioni al mese in ogni biblioteca
- **Assunzione 11:** 40 messaggi al mese in ogni biblioteca
- **Assunzione 12:** alcune consegne comprendono più libri cartacei alla volta

- Assunzione 13: 2 ebook letti da ogni utilizzatore al mese

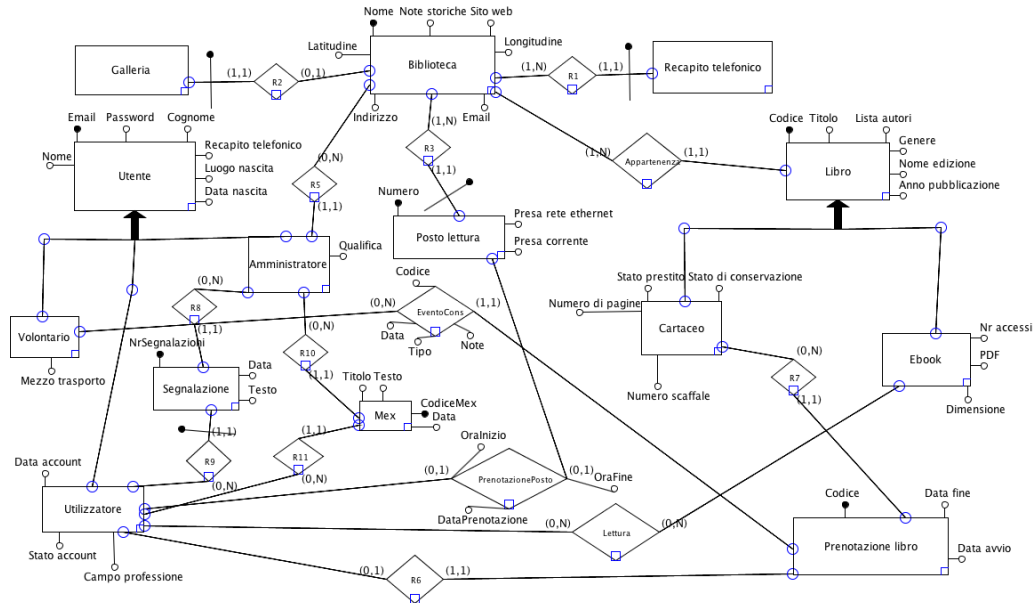
D) GLOSSARIO DEI DATI

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Biblioteca	Luogo dove è possibile prendere in prestito e consultare libri.	Raccolta libri	Recapito telefonico, Galleria, Posto lettura, Amministratore, Libro
Recapito telefonico	Recapito per contattare via telefono la biblioteca.	Numero di telefono	Biblioteca
Galleria immagini	Insieme di immagini che descrivono la biblioteca	Foto	Biblioteca
Utente	Persona che può avere diversi ruoli all'interno della piattaforma EBIBLIO.	Cliente	Utilizzatore, Volontario, Amministratore
Volontario	Utente che porta a termine le consegne dei libri cartacei.	Cliente spontaneo	Prenotazione libro, Utente
Amministratore	Utente che gestisce le operazioni che avvengono sulla piattaforma.	Procuratore	Messaggio, Segnalazione, Biblioteca, Utente
Utilizzatore	Utente che può accedere ai libri ebook e prenotare libri cartacei.	Cliente casuale	Utente, Segnalazione, Posto lettura, Messaggio, Ebook, Prenotazione libro
Posto lettura	Posto che permette agli utilizzatori di stare fisicamente all'interno della biblioteca.	Spazio di lettura	Biblioteca, Utilizzatore
Libro	Oggetto che permette di essere sfogliato e letto, contenuto nella biblioteca.	Opera	Biblioteca, Cartaceo, Ebook
Cartaceo	Libro che si può trovare fisicamente nelle librerie.	Opera fisica	Libro, Prenotazione libro
Ebook	Libro che si trova in versione digitale sul sito della biblioteca.	Opera digitale	Libro, Utilizzatore
Prenotazione libro	Prenotazione obbligatoria richiesta da parte dell'utilizzatore per poter ottenere in prestito un libro cartaceo.	Libro riservato	Cartaceo, Volontario, Utilizzatore
Segnalazione	Avviso da parte di un amministratore destinato a un utente utilizzatore. Serve per poter andare a evidenziare dei comportamenti errati.	Comunicazione	Amministratore, Utilizzatore
Messaggio	Notifica da parte di un amministratore diretto a un destinatario, serve per poter notificare diversi eventi (ex: un libro è diventato prenotabile dopo che un utente l'ha riconsegnato).	Notizia	Amministratore, Utilizzatore

Nel glossario dei dati sono presenti per ogni singolo termine: una breve descrizione, il loro sinonimo e infine che collegamenti hanno verso le altre entità che sono descritte nel modello ER.

2) PROGETTAZIONE CONCETTUALE*

A) DIAGRAMMA ER



B) DIZIONARIO DELL'ENTITA'

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICATORE
Biblioteca	Luogo dove è possibile prendere in prestito e consultare libri.	Nome, Note storiche, Sito web, Longitudine, Latitudine, Indirizzo, <u>Email</u>	Nome
Recapito telefonico	Recapito per contattare via telefono la biblioteca.		NomeBiblioteca
Galleria immagini	Insieme di immagini che descrivono la biblioteca		NomeBiblioteca
Utente	Persona che può avere diversi ruoli all'interno della piattaforma EBIBLIO.	E-mail, Password, Cognome, Nome, Recapito telefonico, Luogo nascita, Data nascita	Email
Volontario	Utente che porta a termine le consegne dei libri cartacei.	Mezzo di trasporto	EmailUtente
Amministratore	Utente che gestisce le operazioni che avvengono sulla piattaforma.	Qualifica	Email, Utente
Utilizzatore	Utente che può accedere ai libri ebook e prenotare libri cartacei.	Data account, Stato account, Campo professione	EmailUtente
<u>Posto lettura</u>	Posto che permette agli utilizzatori di stare fisicamente all'interno della biblioteca.	Numero, presa corrente, presa rete ethernet	Numero, nomeBiblioteca
Libro	Oggetto che permette di essere sfogliato e letto, contenuto nella biblioteca.	Codice, titolo, lista autori, genere, nome edizione, anno pubblicazione	Codice
Cartaceo	Libro che si può trovare fisicamente nelle librerie.	Stato conservazione, stato prestito, numero di pagine, numero scaffale	CodiceLibro
Ebook	Libro che si trova in versione digitale sul sito della biblioteca.	Numero accessi, PDF, Dimensione	CodiceLibro
Prenotazione libro	Prenotazione obbligatoria richiesta da parte dell'utilizzatore per poter ottenere in prestito un libro cartaceo.	Codice, Data fine, Data avvio	Codice
Segnalazione	Avviso da parte di un amministratore destinato a un utente utilizzatore. Serve per poter andare a evidenziare dei comportamenti errati.	Numero segnalazioni, E-mail utilizzatore, Data, Testo	Numero segnalazioni, E-mail utilizzatore
Messaggio	Notifica da parte di un amministratore diretto a un destinatario, serve per poter notificare diversi eventi (ex: un libro è diventato prenotabile dopo che un utente l'ha riconsegnato).	Codice messaggio, Data, Testo, Titolo	Codice messaggio

B) DIZIONARIO DELLE RELAZIONI'

RELAZIONI	DESCRIZIONE	TIPO	COMPONENTI
R1	Relazione che permette di ottenere il recapito telefonico di una biblioteca	1-N	Biblioteca – Recapito telefonico
R2	Relazione che permette di visualizzare immagini relative alla biblioteca che ha una sola galleria di foto.	1-1	Galleria immagini - Biblioteca
R3	Relazione che rappresenta che una biblioteca può avere più posti lettura e che ogni posto lettura è presente all'interno di una biblioteca	1-N	Biblioteca – Posto lettura
Appartenenza	Relazione che descrive in quale biblioteca si trova un determinato libro	1-N	Biblioteca - Libro
R5	Relazione che modella il fatto che una biblioteca può avere più amministratori sulla piattaforma, ma allo stesso tempo un amministratore lavora per una specifica biblioteca	1-N	Biblioteca - Amministratore
R6	Relazione che descrive il fatto che un utente utilizzatore può richiedere o meno una prenotazione per un libro cartaceo e che ogni prenotazione deriva da un singolo utente utilizzatore	1-1	Utilizzatore – Prenotazione libro
R7	Relazione che specifica il fatto che un libro cartaceo può avere molte richieste di prenotazione e una prenotazione viene richiesta ad un singolo libro.	1-N	Cartaceo – Prenotazione libro
R8	Relazione che descrive il fatto che un amministratore può inviare diverse segnalazioni, e che ogni segnalazione viene inviata da un amministratore.	1-N	Amministratore-Segnalazione
R9	Relazione che dimostra il fatto che un utilizzatore può ricevere diverse segnalazioni e allo stesso tempo ogni segnalazione è riferita a un utente utilizzatore.	1-N	Segnalazione-Utilizzatore
R10	Relazione che permette all'amministratore di avere la facoltà di inviare più messaggi all'utilizzatore e che ogni messaggio viene mandato da un amministratore.	1-N	Amministratore-Messaggio
R11	Relazione che indica che un messaggio fa riferimento a un singolo utilizzatore e che egli può ricevere più di un messaggio.	1-N	Messaggio-Utilizzatore
Lettura	Relazione che permette all'utente utilizzatore di poter accedere a diversi ebook presenti sulla piattaforma, e allo stesso tempo un ebook può essere acceduto da diversi utilizzatori	N-N	Utilizzatore - Ebook
Prenotazione posto	Relazione la quale permette all'utente utilizzatore di avere la possibilità di prenotare un posto lettura all'interno della biblioteca ed esso può essere occupato da un singolo utilizzatore	1-1	Utilizzatore – Posto lettura

C) BUSINESS RULES E REGOLE DI DERIVAZIONE

REGOLE DI DERIVAZIONE
La prenotazione libro da parte di un utente utilizzatore può avvenire se Stato prestito = "Disponibile" e Stato di conservazione != "Scadente"
Se un utente ha Segnalazione>3 allora il suo stato account = "Sospeso" (e non può più accedere alla piattaforma)
La prenotazione posto è possibile qualora la biblioteca abbia posti lettura disponibile per quell'ora
La data di fine della prenotazione libro = data inizio + 15 giorni
Gli utilizzatori possono accedere liberamente agli e-book disponibili: tuttavia, il sistema tiene traccia dello storico degli accessi agli e-book (o meglio alle loro schede) effettuati da ciascun utente.

REGOLE DI VINCOLO
Lo stato di conservazione può essere: "Ottimo", "Buono", "Non Buono", "Scadente"
o stato prestito può essere: "Disponibile", "Prenotato", "Consegnato"
La qualifica degli utenti amministratori può essere di massimo 10 caratteri
Il mezzo di trasporto degli utenti volontari può essere: "piedi", "bici", "auto"
Lo stato dell'account può essere: "Attivo", "Sospeso"
Il tipo di un evento di consegna può essere: "Restituzione", "Affidamento"

3) PROGETTAZIONE LOGICA

A) VALUTAZIONE RIDONDANZA RIDONDANZE

COSTANTI

- α (peso operazioni scrittura) = 2
- wI (peso operazioni interattive) = 1
- wB (peso operazioni batch) = 0.5

ANALISI DEI COSTI CON RIDONDANZA (Numero accessi presente in Ebook)

1) Inserire un nuovo e-book (Interattiva, 3 volta/mese)

$$C(op1) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(op1) = 3 * 1 * (2 * 2 + 0) = 12$$

Il numero di operazioni in scrittura è pari a 2 per inserire un nuovo Ebook vi è un accesso all'entità Ebook e, successivamente, nella relazione Lettura che collega Ebook con Utilizzatore per aggiungere il nuovo Ebook .

2) Rimuovere un Ebook (Interattiva, 3 volta/mese)

$$C(op2) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(op2) = 3 * 1 * (2 * 2 + 0) = 12$$

Il numero di operazioni in scrittura è pari a 2 poichè viene eliminata dall'entità Ebook e si deve anche accedere alla relazione Lettura per eliminare se ci sono delle coppie Utilizzatore-Ebook.

3) Contare numero accessi di ogni Ebook (Interattiva, 2 volta/mese)

$$C(op3) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(op3) = 2 * 1 * (2 * 0 + 1) = 2$$

In questo caso si parla di operazioni di count, perciò il numero di operazioni di lettura è pari a 1 perchè per vedere il numero di accessi in ogni Ebook si accede direttamente all'entità in questione grazie all'attributo NrAccessi.

COSTO TOTALE

$$C(\text{op_rid}) = C(\text{op1}) + C(\text{op2}) + C(\text{op3}) + C(\text{op4}) = 12 + 12 + 2 = 26$$

Il costo totale delle operazioni è pari a 26

ANALISI DEI COSTI SENZA RIDONDANZA (Numero accessi NON presente in Ebook)

1)Inserire un nuovo e-book (Interattiva, 3 volta/mese)

$$C(\text{op1}) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(\text{op1}) = 3 * 1 * (2 * 2 + 0) = 12$$

Il costo di questa operazione non cambia anche rimuovendo la ridondanza.

2)Rimuovere un Ebook (Interattiva, 3 volta/mese)

$$C(\text{op2}) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(\text{op2}) = 3 * 1 * (2 * 2 + 0) = 12$$

Il costo di questa operazione non cambia anche rimuovendo la ridondanza.

3)Contare numero accessi di ogni Ebook (Interattiva, 2 volta/mese)

$$C(\text{op3}) = f(o_t) * w_t * (\alpha * NC_write + NC_read)$$

$$C(\text{op3}) = 2 * 1 * (2 * 0 + 5000) = 10\ 000$$

Il costo dell'operazione è pari a 10 000, che è molto più grande rispetto al costo che ha mantenere la ridondanza (2). Questo perchè per contare il NrAccessi bisogna far riferimento alla tabella dei volumi poichè definisce che all'interno della relazione lettura vi sono 5000 coppie, perciò si accede a quella relazione e si contano il numero di accessi.

COSTO TOTALE

$$C(\text{op}) = C(\text{op1}) + C(\text{op2}) + C(\text{op3}) + C(\text{op4}) = 12 + 12 + 10\ 000 = 10\ 024$$

Il costo totale delle operazioni è passato da 26 a 10 024, un incremento esponenziale.

VALUTAZIONE SULLA RIDONDANZA

Per poter andare a definire se è meglio tenere o meno la ridondanza "NrAccessi" si calcolano i seguenti indici.

SPEED UP

$$C(\text{op}) / C(\text{op_rid}) = 10\ 024 / 26 = 385,54$$

Questo indice si calcola facendo il rapporto tra il totale del costo delle operazioni senza ridondanza (C(op)) e il costo totale con l'attributo ridondante (C(op_rid)), e permette di vedere se conviene mantenere o meno la ridondanza.

Se l'attributo ha valore circa 1 significa che quasi indifferente avere o meno la ridondanza; se maggiore di 1, come in questo caso, significa che la ridondanza abbate il costo e migliora, quindi, il database; infine, se minore di 1, l'attributo aggiunto aumenta il costo e dunque la ridondanza è da evitare.

Il primo indice di speed-up dice che la ridondanza va tenuta poichè il valore che abbiamo ottenuto (385,54) è molto alto. Non basta, però, l'indice di Speed-up per decretare se mantenere o no una ridondanza, infatti va calcolato e confrontato anche la memoria occupata dall'attributo "NrAccessi" all'entità "Ebook".

FATTORE MEMORIA

$$m(S) = X$$

$$m(S) = X + 15\ 000 * 4 = X + 60\ 000(\text{Byte})$$

Il valore di 60 000 Byte si ottiene aggiungendo un intero (numeroAccessi) che equivale a 4 Byte e moltiplicando l'intero per tutte le volte che si trova nel database, in questo cas in tutti gli Ebook che è stato assunto essere presenti nel sistema.

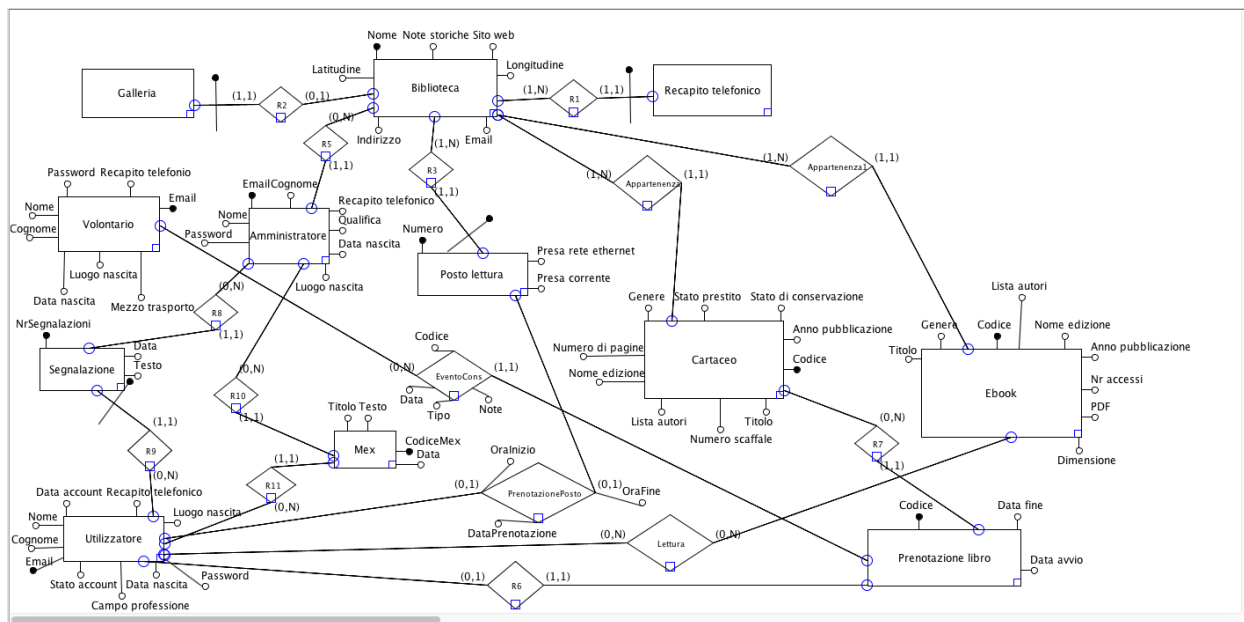
Dalla tabella dei volumi si evidenzia che si hanno 15 000 “Ebook”, perciò viene aggiunto 15 000 campi nuovi dove ogni campo pesa 4 Byte.

$$|m(S) - m(S_rid)| = |X - X - 60\,000| = 60\,000 \text{ Byte}$$

In conclusione, posso affermare che conviene mantenere la ridondanza “numeroAccessi” poichè l’attributo occupa una quantità di memoria non troppo alta (60 000 Byte), ma allo stesso tempo riduce notevolmente il costo totale delle operazioni da 10 024 a 26.

B) RISTRUTTURAZIONE DELLO SCHEMA CONCETTUALE

Per tradurre le generalizzazioni è stato adottato il modello di traduzione delle generalizzazioni dove viene rimossa l’entità padre trasferendo tutti gli attributi e le relazioni nell’entità figlie. Nel diagramma si può anche notare che si è creata anche la relazione “Appartenenza1” che è identica ad “Appartenenza” e serve per poter collegare Biblioteca con le entità figlie “Ebook” e “Cartaceo”.



TRADUZIONE IN TABELLE DEL MODELLO ER APPLICANDO TRADUZIONE DELLE GENERALIZZAZIONI MODELLO 2

- **FRANCOBOLLO** (Numero francobollo, paese di emissione, dimensione, anno)
- **BIBLIOTECA** (Nome, NoteStoriche, SitoWeb, Longitudine, Latitudine, Indirizzo, E-mail)
- **GALLERIA** (NomeBiblioteca)
- **RECAPITO TELEFONICO** (NomeBiblioteca)
- **CARTACEO** (Codice, Titolo, ListaAutori, Genere, NomeEdizione, AnnoPubblicazione, StatoConservazione, StatoPrestito, NrPagine, NumeroScaffale, NomeBiblioteca)
- **EBOOK** (Codice, Titolo, ListaAutori, Genere, NomeEdizione, AnnoPubblicazione, PDF, NrAccessi, Dimensione, NomeBiblioteca)
- **POSTO LETTURA** (Numero, NomeBiblioteca PresaCorrente, PresaReteEthernet)
- **PRENOTAZIONEPOSTO** (NomeBibliotecaPostoLettura, EmailUtilizzatore, DataPrenotazione, OraFine, OraInizio, NumeroPostoLettura)
- **LETTURA** (CodiceEbook, EmailUtilizzatore)

- **PRENOTAZIONELIBRO** (Codice, DataFine, DataAvvio, CodiceCartaceo, E-mailUtilizzatore)
- **EVENTOCONSEGNA** (Codice, CodicePrenotazioneLibro, E-mailVolontario, Data, Tipo, Note)
- **AMMINISTRATORE** (Email, Password, Cognome, Nome, LuogoNascita, DataNascita, RecapitoTelefonico, Qualifica, NomeBiblioteca)
- **VOLONTARIO** (Email, Password, Cognome, Nome, LuogoNascita, DataNascita, RecapitoTelefonico, MezzoDiTrasporto)
- **UTILIZZATORE** (Email, Password, Cognome, Nome, LuogoNascita, DataNascita, RecapitoTelefonico, DataAccount, StatoAccount, CampoProfessione)
- **SEGNALAZIONE** (EmailUtilizzatore, NrSegnalazioni Data, Testo, E-mailAmministratore)
- **MESSAGGIO** (Codice, Titolo, Testo, Data, E-mailAmministratore, E-mailUtilizzatore)

VINCOLI DI INTEGRITA' REFERENZIALE

- **GALLERIA.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **RECAPITOTELEFONICO.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **CARTACEO.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **EBOOK.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **POSTOLETTURA.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **PRENOTAZIONEPOSTO.NumeroPostoLettura** -> **POSTOLETTURA.Numero**
- **PRENOTAZIONEPOSTO.NomeBiblioteca** -> **POSTOLETTURA.NomeBiblioteca**
- **PRENOTAZIONEPOSTO.EmailUtilizzatore** -> **UTILIZZATORE.Email**
- **LETTURA.CodiceEbook** -> **EBOOK.Codice**
- **LETTURA.EmailUtilizzatore** -> **UTILIZZATORE.Email**
- **PRENOTAZIONELIBRO.CodiceCartaceo** -> **CARTACEO.Codice**
- **PRENOTAZIONELIBRO.EmailUtilizzatore** -> **UTILIZZATORE.Email**
- **EVENTOCONSEGNA.CodicePrenotazioneLibro** -> **PRENOTAZIONELIBRO.Codice**
- **EVENTOCONSEGNA.EmailVolontario** -> **VOLONTARIO.Email**
- **AMMINISTRATORE.NomeBiblioteca** -> **BIBLIOTECA.Nome**
- **SEGNALAZIONE.EmailUtilizzatore** -> **UTILIZZATORE.Email**
- **SEGNALAZIONE.EmailAmministratore** -> **AMMINISTRATORE.Email**
- **MESSAGGIO.EmailUtilizzatore** -> **UTILIZZATORE.Email**
- **MESSAGGIO.EmailAmministratore** -> **AMMINISTRATORE.Email**

DESCRIZIONE AD ALTO LIVELLO DELLE FUNZIONALITA' DELL'APPLICAZIONE WEB

L'utente interagisce inizialmente con il file **login.php**, dove è presente la possibilità di accedere o creare un account può accedere ed entrare, poichè è possibile utilizzare il portale solo con un account.

All'interno di questa schermata si ha la possibilità di poter compiere diverse azioni:

- Per prima cosa un utente già registrato al sito può inserire il mio username e la mia password, specificando che tipologia di utente sono: utilizzatore (utente), volontario, amministratore e procedere con il processo di autenticazione. Se le credenziali sono corrette e la tipologia di account selezionata corrisponde al proprio ruolo allora si accederà direttamente alla propria pagina con le diverse azioni possibili. Se invece le credenziali sono state inserite con degli errori, oppure il proprio account è stato sospeso (un account viene sospeso quando riceve più di 3 segnalazioni), allora apparirà un messaggio di errore evidenziando il fatto che l'accesso è stato negato
- Nel caso in cui un utente si affacci per la prima volta a questa piattaforma vi è la possibilità di registrazione come utilizzatore o come volontario. L'operazione è molto semplice: sarà presente un link che porta direttamente alla pagina di registrazione dove, una volta compilati i campi richiesti, sarà possibile completare la creazione dell'account. Un account non viene creato nel caso in cui venga inserita un email già in possesso da uno qualsiasi degli utenti che hanno una registrazione nel sito (sia utilizzatore, che volontari o amministratore). Questo controllo è fatto per far sì che ogni singola email sia univoca all'interno della biblioteca anche tra ruoli diversi.
- L'unica azione che può compiere un utente ancora prima ancora di effettuare la registrazione, si attiva cliccando nella barra in alto a sinistra nella voce **“Informazioni”** oppure **“Statistiche”**, e si tratta dell'indirizzamento alla pagina che fornisce diverse nozioni interessanti sulle biblioteche e sulle statistiche del portale.

DESCRIZIONE PAGINE DEGLI UTENTI

Una volta che ha effettuato l'accesso con successo, un utente accede alla pagina relativa alla propria tipologia di account, dove ognuna ha diverse operazioni possibili:

- **Utilizzatori** = Gli utilizzatori hanno la possibilità di prenotare posti lettura e libri (all'interno della pagina **“Utilizzatore.php”**), di poter accedere a qualsiasi informazioni su un libro cartaceo e su un Ebook e la possibilità ricevere messaggi o segnalazioni da parte degli amministratori di una biblioteca.
- **Volontari** = I volontari si occupano di consegnare i libri cartacei agli utenti utilizzatore e poi riportarli, una volta scaduto il prestito, nella biblioteca. E' proprio nella pagina **“Volontario.php”** che possono inserire, aggiornare o eliminare un evento consegna e vedere quali consegne sono state effettuate e quali ancora devono essere portate a termine.
- **Amministratori** = Gli amministratori (la cui pagina di riferimento è **“Amministratore.php”**) hanno il compito di sorvegliare ed effettuare operazioni mirante al controllo della biblioteca di cui hanno la gestione in carico. Possono aggiungere, rimuovere o aggiornare libri cartacei di una biblioteca, avere il controllo su ogni prenotazione fatta ad un posto o un libro nella loro biblioteca ed infine hanno il compito di inviare messaggi agli utenti per avvisi particolari, ma anche di segnalare persone che compiono azioni sbagliate o commettono degli errori (ex: trattare male un libro o ritardare di molto la consegna del libro dopo la data di scadenza massima prefissata). Nel caso possono anche togliere la sospensione di un account rimuovendo da esso tutte le segnalazioni inviate in precedenza.

In ogni pagina una volta loggati si può facilmente uscire cliccando sul bottone in alto a destra **“Logout”** che farà tornare l'utente alla pagina iniziale **login.php**.

AUTENTICAZIONE PIATTAFORMA

Ebiblio Informazioni Statistiche Accedi/Registri

Benvenuto su Ebiblio

☐ Utente ☐ Volontario ☐ Admin

Non hai un account? [Registrati come Utente](#)

Oppure: [Registrati come Volontario](#)

Data la presenza di tre diversi tipi di utente si è scelto di realizzare un unico log-in condiviso, dove il tipo di account viene selezionato attraverso un radio-button.

In base alla scelta effettuata nel radio botton, viene assegnata alla variabile “\$account” una stringa che descrive il tipo di account.

Successivamente viene controllato il valore di questa attraverso degli if annidati.

Nel caso si sia scelto un account di tipo utilizzatore, viene eseguita una query in cui viene selezionata l’email e la password di un account in cui l’email corrisponde a quella inserita. Inoltre, si controlla che lo stato dell’account non sia stato sospeso. Lo stesso vale per gli altri tipi di account, a differenza che gli account di tipo volontario e admin non abbiano attributi che definiscono lo stato di questi.

Infine, viene controllata se la password inserita coincide con la password ritornata dalla query.

```

if($account == "Utilizzatore"){
    $sql = "SELECT email, Psw FROM utilizzatore WHERE email = ? AND StatoAccount='Attivo'";
}
else if($account == "Volontario"){
    $sql = "SELECT email, Psw FROM volontario WHERE email = ?";
}
else if($account == "Amministratore"){
    $sql = "SELECT email, Psw FROM amministratore WHERE email = ?";
}

```

GESTIONE DEGLI ERRORI

```

$email = $password = "";
$email_err = $password_err = "";
$account_err = $account = "";

```

Per gestire correttamente l’autenticazione vengono utilizzate delle variabili php per gestire il contenuto dell’inputBox e gli eventuali errori.

Queste variabili, inizialmente, sono settate vuote, nel caso avvengano errori, il contenuto cambierà e verranno visualizzati a video (una spiegazione più approfondita viene specificata in seguito).

Nel caso non venga selezionato nessuno dei radio-button verrà visualizzata una “alert-box” dove viene stampato il messaggio di mancata selezione del tipo di account.

Invece nel caso che le credenziali siano sbagliate viene visualizzata una nuova “alert-Box”:

Nel caso invece, non vengano inseriti dati nel inputBox, grazie agli attributi required e un controllo php sui dati passati tramite il metodo POST assicura che le stringhe passate non siano vuote e che sia inserita una email.

```
if(empty(trim($_POST["email"]))) {
    $email_err = "Inserisci l'email.";
} else {
    $email = trim($_POST["email"]);
}
```

```
if(empty(trim($_POST["password"]))) {
    $password_err = "Inserisci la password.";
} else {
    $password = trim($_POST["password"]);
}
```

```
if(!isset($_POST["account"])) {
    $account_err = "Inserisci il tipo del tuo account";
} else {
    $account = trim($_POST["account"]);
}
```

Invece nel caso di una corretta autenticazione vengono assegnati a 3 diverse variabili di sessioni i seguenti valori:


```
$_SESSION["loggedin"] = true;
$_SESSION["email"] = $email;
$_SESSION['account'] = $_POST["account"];
```

La prima indica se si è autenticati o meno, la seconda l'email dell'utente e nell'ultima il tipo di account con cui si è fatto l'accesso ("Utente", "Volontario", "Admin").

Abbiamo scelto di utilizzare delle variabili di sessione per mantenere le credenziali dell'utente durante l'utilizzo della piattaforma per facilitare la navigazione. Nel caso un utente chiuda la pagina, le variabili continueranno a mantenere il valore finché l'utente non chiuderà il browser quando il valore delle variabili verrà "un-settato".

REGISTRAZIONE

Dopo un primo tentativo di realizzare una pagina unica per la registrazione di account di tipo Utente e di tipo Volontario, abbiamo deciso di separarla in due pagine differenti ("registerUser.php" e "registerVol") per facilitare e semplificare i diversi due approcci dato che differenziavano di alcuni campi.

Questo è il form della registrazione utente:

Il funzionamento e la gestione degli errori è simile alla pagina login.php

Questo è il form della registrazione utente:

Inserisci i tuoi dati

Email

Password

Nome

Cognome

Luogo di nascita:

Data di nascita:

telefono:

Professione:

Registrati

FILE = INDEX.PHP

Il file **index.php**, attraverso le diverse variabili di sessione, si occupa di indirizzare i diversi utenti nella “homepage” del proprio tipo di account (utente, volontario, admin) e nel caso l’utente non abbia ancora effettuato l’accesso viene reindirizzato nella pagina del login/registrazione.

```

if ( isset($_SESSION['account'])) {
    if ($_SESSION['account'] == 'Utilizzatore'){
        header("location: Utilizzatore.php");
        exit;
    }
    else if ($_SESSION['account'] == 'Volontario'){
        header("location: Volontario.php");
        exit;
    }

    else if ($_SESSION['account'] == 'Amministratore'){
        header("location:Amministratore.php");
        exit;
    }
}
else
{
    header("location:login.php");
}
}

```

BOOTSTRAP / GRAFICA:

L'utilizzo dei modelli di Bootstrap viene reso disponibile grazie a questa dichiarazione:

```

<title>Login</title>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
<link rel="stylesheet" href="css/style.css">

integrity="sha384-Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
integrity="sha384-KJ3o2DKtIkVYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>

```

E nella definizione degli oggetti HTML di una speciale classe che “punteranno” ai script precedentemente dichiarati:

```

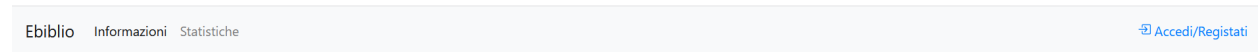
<input name="email" class="form-control" placeholder="Email" type="email" value="<?php echo $email; ?>" required>

```

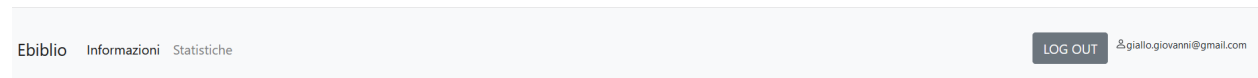
BARRA DI NAVIGAZIONE

La barra di navigazione viene ‘implementata’ in ogni pagina, offre i collegamenti a Home (Ebiblio), Informazioni e Statistiche.

Nel caso un utente non abbia effettuato l’accesso, le diverse variabili di sessione non saranno impostate, verrà visualizzato un collegamento della pagina di login e inoltre un oggetto bootstrap che indica la possibilità di autenticazione.



Nel caso un account effettui un accesso, la barra si modifica in questo modo:



Viene visualizzata l’email dell’account che ha effettuato l’accesso e viene inserito un bottone per il logout, l’email dell’account con il quale si è fatto l’accesso e anche un oggetto bootstrap raffigurante un profilo. Questa è l’implementazione:

```
<?php
    if (!isset($_SESSION["loggedin"]) && !isset($_SESSION["email"])){ ?>
        <div class="nav-registersection">
            <li><a href="login.php"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentCol
            <path fill-rule="evenodd" d="M6 3.5a.5.5 0 0 1 .5-.5h8a.5.5 0 0 1 .5.5v9a.5.5 0 0 1-.5.5h-8a.5.5 0 0 1-
            <path fill-rule="evenodd" d="M11.854 8.354a.5.5 0 0 0-.708l-3-3a.5.5 0 1 0-.708.708L10.293 7.5H1.5a.5
        </svg> Accedi/Registati</a></li>
        </div>
    <?php }
    else{
        ?>
    </li>
    <div class="namelog" id="logName">
        <svg xmlns="http://www.w3.org/2000/svg" width="14" height="14" fill="currentColor" class="bi bi-person"
        <path d="M8 8a3 3 0 1 0 0-6 3 3 0 0 0 6zm2-3a2 2 0 1 1-4 0 2 2 0 0 1 4 0zm4 8c0 1-1 1-1 1H3s-1 0-1-1 1-
        </svg><?php echo $_SESSION['email']; ?>
    </li>
</div>
<?php } ?>
```

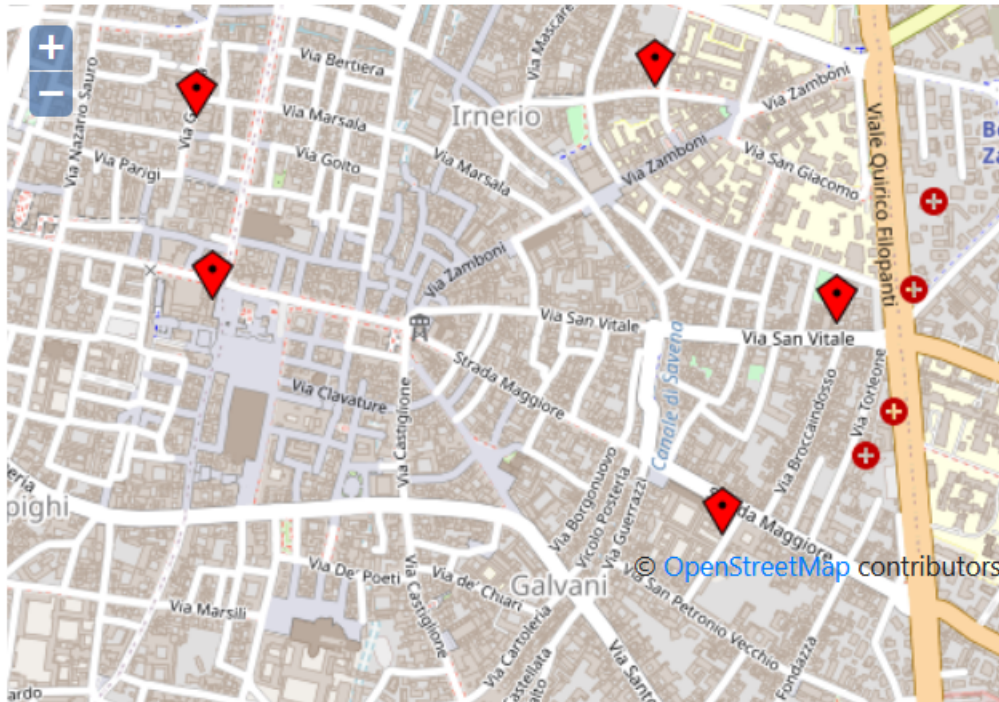
FUNZIONE LOG-OUT

Nel caso venga premuto il bottone per il log-out, le variabili di sessioni vengono “un-settate” e si viene re-indirizzati nella pagina di login:

```
function logOut()
{
    session_unset();
    header("location: login.php");
    exit;
}
if(array_key_exists('logout',$_POST)){
    logout();
}
```

IMPLEMENTAZIONE DELLA MAPPA DELLE BIBLIOTECHE

In **Informazioni/Visualizza Informazione delle biblioteche**, in basso, è possibile visualizzare una mappa in cui sono indicate le diverse biblioteche che fanno parte di Ebiblio:



Abbiamo implementato questa mappa grazie all'utilizzo di OpenLayers, nella pagina delle informazioni viene dichiarato questo script:

```
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
```

Per poter visualizzare la mappa, inizialmente viene dichiarato un oggetto mappa e le coordinate di Longitudine/Latitudine per indicare il centro della mappa:

```
// Definisco la variabile mappa come un oggetto OpenLayers.Map utilizzando il DivMappa, poi aggiungo il Layer OSM (Open Street Map)
var Mappa = new OpenLayers.Map("DivMappa");
Mappa.addLayer(new OpenLayers.Layer.OSM());
// Creo l'oggetto contenente le coordinate (prima longitudine e poi latitudine)
var LonLat = new OpenLayers.LonLat(11.356225465875625, 44.493432020547615)
    .transform(
        new OpenLayers.Projection("EPSG:4326"), // Trasformazione da WGS 1984..
        Mappa.getProjectionObject() // .. a Spherical Mercator Projection
    );
```

Successivamente viene settato uno zoom:

```
// Imposto lo zoom
var zoom=16;
```

E viene centrata:

```
// Imposto le coordinate di lonLat come centro della mappa di partenza
Mappa.setCenter (LonLat, zoom);
```

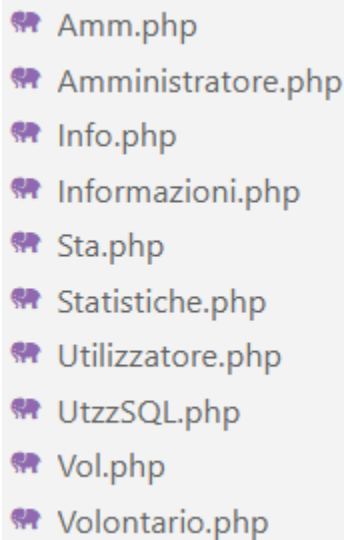
I **markers**, invece, vengono dichiarati come il punto “LonLat” spiegato sopra.

E Infine aggiunti alla mappa in questo modo:

```
// Creo una variabile contenente il layer dei marker poi collego il layer dei markers alla mappa
var LayerMarkers = new OpenLayers.Layer.Markers( "Markers" );
Mappa.addLayer(LayerMarkers);
// Aggiungo al layer dei marker un marker (utilizzando l'oggetto LonLat per le coordinate)
LayerMarkers.addMarker(new OpenLayers.Marker(LonLat2));
```

PHP

La prima operazione da attuare su PHP è la connessione al database che permette di interagire con il database attraverso stored procedure o, direttamente, query su PHP. La connessione per il database ‘EBIBLIO’ avviene insieme agli input dell’host della connessione e la password ‘root’ del database stesso.



Una volta collegato il database su PHP, inizia la progettazione della struttura HTML su cui lanciare le stored procedure.

```
$db = mysqli_connect('localhost','root','','EBIBLIO') or die("Non è stato possibile connetersi al DB");
```

Si è optato per la creazione di una pagina con la parte di struttura HTML che passasse un input alla pagina PHP con l’operazione sui dati da compiere e gli input per la stessa.

Vi è quindi una pagina PHP alla quale si accede attraverso il browser con il codice HTML collegata ad una pagina, sempre PHP, che gestisce, invece, il lancio delle stored procedure.

Per avere un codice ordinato e semplice da modificare si è scelto di creare un file PHP con il lancio delle stored per ogni pagina HTML, quindi esiste una pagina PHP singola per Amministratore, Utilizzatore, Volontario, Statistiche, Informazioni (tutte le pagine web da cui si possono usare delle stored). Ad esempio, alla pagina con la parte HTML ‘Amministratore.PHP’ è collegata al file PHP ‘Amm.php’.

Il file ‘Amministratore.php’ è strutturato con un form HTML per ogni stored che restituisce al file ‘Amm.php’ degli input attraverso il metodo “post”.

“Post” è uno dei possibili metodi con cui si può gestire un input su HTML e si basa sulla creazione di un form di questo tipo e la successiva dichiarazione di variabili che poi potranno essere ricevute in un altro file attraverso vari comandi, tra i quali è stato scelto: “\$_REQUEST[‘nomevariabile’]”.

```
<form action = "Amm.php" method = "post">  
|
```

Gli input restituiti mediante il metodo spiegato precedentemente sono di due tipi:

- Il numero della stored da lanciare;
- I dati che servono in input alla procedure.

Il primo input è fondamentale per come è gestito il lancio delle stored procedure, infatti ogni volta che l’utente sceglie una di queste sulla pagina del browser, attiva contemporaneamente un input di tipo ‘hidden’ che invia il numero della operazione che è stata selezionata. L’input ‘hidden’ (nascosto) permette di inviare un dato in input senza che si veda nella pagina web lato front-end ma solo nel codice, in questo modo l’utente manda il numero della stored che vuole lanciare in modo automatico, senza vederlo.

```
<input type = "hidden" name = "numero" id = "numero" value = "1">
```

Inviato l’input, sulla pagina di ‘Amm.php’ il numero viene assegnato ad una variabile per poi, attraverso un controllo ‘if’, entrare nella parte di codice della rispettiva procedure.

```
$numero = $_REQUEST['numero'];  
//-----  
if($numero == 1){  
|
```

Il secondo tipo di input, invece, viene inserito dall’utente dalla pagina web e poi, sempre mediante il comando “\$_REQUEST”, viene assegnato ad una variabile. Nel caso dell’input di una stringa viene usato il comando ‘mysqli_real_escape_string’ per permettere una lettura dei caratteri speciali.

Si nota come per il dato relativo alla mail viene usato un comando diverso. ‘\$_SESSION[‘email’]’ restituisce il valore di una variabile di sistema, in questo caso la mail, che è salvata al momento del login. L’utilizzo della variabile di sistema permette all’utente di non scrivere ogni volta la mail per la lanciare un’operazione.

```
$codice3 = $_REQUEST['inputCodice3'];  
$titolo3 = mysqli_real_escape_string($db, $_REQUEST['inputTitolo3']);  
$anno3 = $_REQUEST['inputAnno3'];  
$stato3 = mysqli_real_escape_string($db, $_REQUEST['inputStato3']);  
$pagine3 = $_REQUEST['inputPagine3'];  
$biblioteca3 = mysqli_real_escape_string($db, $_REQUEST['inputBiblioteca3']);
```

Assegnati i dati inseriti dall’utente alle variabili, queste vengono messe nelle righe di codice dove vi è la chiamata in linguaggio mysql della stored procedure.

```
$MM_query = "CALL MandaMessaggio('$titolo5','$testo5','$data5','$email5','$emailU5')";
```

Il risultato della procedure viene assegnato ad una variabile per poter gestire l’output che restituisce. L’output può considerarsi di due tipologie: derivato da una operazione di reading o di writing.

Nelle operazioni di reading vengono restituite delle righe che vengono stampate attraverso un ciclo foreach. In quelle di writing, invece, non viene restituito niente come output e quindi viene stampato semplicemente una frase per indicare all'utente che l'operazione è riuscita.

Per controllare il funzionamento della stored viene usato il comando 'mysqli_affected_rows(database)' che restituisce il numero di righe che hanno subito un interessamento da parte dell'ultima operazione attuata nel database.

Qualora il risultato di questo comando PHP fosse maggiore di 0, si può assodare che la stored ha avuto successo poiché sono state modificate o selezionate delle righe; nel caso in cui venga restituito 0, invece, significa che la procedure non ha sortito alcun effetto sul database e quindi verosimilmente non ha funzionato come avrebbe dovuto.

Potrebbe anche restituire -1 nel caso in cui l'operazione restituisca un errore in MySQL.

Poiché sia l'errore MySQL sia il caso in cui la stored non compia alcun effetto sulle righe dipendono da l'inserimento sbagliato di dati dalla pagina web, si è deciso di gestire con una frase unica di errore entrambi.

```
$varCount5 = mysqli_affected_rows($db);
if ($varCount5<=0) {
    echo("<br>");
    echo("Non è stato possibile inviare il messaggio");
}
else {
    echo("<br>");
    echo("Il messaggio è stato inviato con successo");
}
```

Infine, la pagina PHP si conclude con un link che riconduce alla pagina HTML corrispondente.

Ad esempio 'Amm.PHP' avrà il link per la pagina 'Amministratore.PHP'.

MONGODB

Ad aggiungersi al database in linguaggio MYSQL spiegato nelle sue funzioni e interazioni prima, vi è un compito della gestione del sistema che è compiuto da un database in linguaggio NOSQL, in particolare MONGODB.

Questo database si popola solo con stringhe inserite dalla piattaforma quando avviene un'operazione di INSERT nella database principale, tanto da dare un ruolo alla parte NOSQL da 'log' che tiene traccia delle operazioni. Sotto alla procedure con un 'INSERT' da effettuare, nel file PHP dedito all'esecuzione delle procedure è presente la parte di codice che prima si connette al database 'MONGODB' e poi compie un inserimento di una stringa che spiega in modo generale l'operazione 'INSERT' sul database MYSQL.

```
$bulk = new MongoDB\Driver\BulkWrite;
$doc = [ 'log' => 'Inserito un messaggio da '.$email5 ];
$bulk->insert($doc);
$result = $manager->executeBulkWrite('Ebiblio.Log', $bulk);
```

Ad esempio, nel caso illustrato la stringa indica quale elemento sia stato inserito (un cartaceo) e che utente l'abbia fatto ('\$email1', ovvero l'e-mail dell'Amministratore).

STORED PROCEDURE

PRENOTA_POSTO


```

CREATE PROCEDURE PrenotaPosto(IN NomeB VARCHAR(100), IN EmailU VARCHAR(100), IN Dat DATE,
                               IN OraI TIME, IN OraF TIME, IN Corrente BOOLEAN, IN Ethernet BOOLEAN)
BEGIN
    DECLARE TIM TIME;
    DECLARE X INT;
    DECLARE I INT;
    DECLARE Fine TIME;
    DECLARE Inizio TIME;
    DECLARE Nr INT;
    DECLARE Posto INT;
    SET TIM = TIMEDIFF(OraF, OraI);
    SET X = ((TIME_TO_SEC(TIM)/60)/60);
    SET I = 1;
    SET Fine = OraF;
    SET Inizio = OraI;
    SET Posto = (SELECT Numero FROM POSTOLETTURA
                  WHERE (NomeBiblioteca=NomeB) AND (PresCorrente=Corrente) AND (PresEthernet=Ethernet) AND Numero NOT IN
                  (SELECT NumeroPostoLetture FROM PRENOTAZIONEPOSTO WHERE (OraI<OraFine) AND NumeroPostoLetture IN
                  (SELECT NumeroPostoLetture FROM PRENOTAZIONEPOSTO
                   WHERE (NomeBibliotecaPostoLetture=NomeB) AND (OraF>OraInizio) AND (OraF<=OraFine) AND (Dat=DataPrenotazione)))
                  ORDER BY Numero LIMIT 1);
    WHILE (I<=X) DO
        SET Fine = ADDTIME(Fine, '01:00:00');
        INSERT INTO PRENOTAZIONEPOSTO(NomeBibliotecaPostoLetture, NumeroPostoLetture, EmailUtilizzatore, DataPrenotazione, OraInizio, OraFine)
        VALUES(NomeB, Posto, EmailU, Dat, Inizio, Fine);
        SET I = I+1;
        SET Inizio = ADDTIME(Inizio, '01:00:00');
    END WHILE;

```

PrenotaPosto è la stored procedure che permette ad un utente utilizzatore di prenotare un posto lettura in una biblioteca. L'utente('EmailU') indica in quale biblioteca ('NomeB'), in che giorno('Dat'), a che ora ('OraI' e 'OraF') e quali caratteristiche ('Corrente' e 'Ethernet') debba avere il posto lettura che vuole prenotare. Forniti tutti gli input, la procedure deve fare numerosi passaggi per finire con l'operazione di 'INSERT' nella tabella PRENOTAZIONEPOSTO.

La variabile Posto è il numero della postazione all'interno della biblioteca che l'utente prenoterà; la sua individuazione rappresenta il primo passaggio della procedure. Dal momento che l'unica caratteristica che differenzia un posto lettura dall'altro è la presenza della presa di corrente e del cavo ethernet, si presuppone che un numero di posto rispetto ad un altro sia uguale per l'utente, quindi il numero viene scelto dal sistema prendendo il primo numero in ordine crescente tra le varie postazioni libere che rispettano tutte le caratteristiche date in input dall'utente.

L'altro nodo da sciogliere prima di compiere l'inserimento è: l'orario. Nella gestione dell'ora di inizio e di fine si è cercato di mettere meno vincoli possibili nella scelta da parte dell'utilizzatore ma allo stesso tempo di poter gestire nel modo più efficiente la tabella PRENOTAZIONEPOSTO. In concreto, per la gestione del sistema si è scelto di catalogare ogni prenotazione da un'ora in modo da rendere ogni prenotazione uguale ed evitare di dover ponderare il peso di prenotazioni più lunghe nelle statistiche e anche per aiutare la ricerca e le eliminazioni delle righe di PRENOTAZIONEPOSTO.

Invece all'utilizzatore è permesso prenotare per più ore di fila con il solo vincolo che siano orari spaccati (es. 9:00, 11:00), infatti sarà la procedure poi a dividere in più prenotazioni da un'ora. Alla variabile 'X' assegno la differenza tra l'ora di fine prenotazione e quella di inizio trasformata in un intero, ovvero il numero di ore prenotate.

In base a quante ore sono state prenotate farò un numero corrispondente di operazioni di 'INSERT' di prenotazioni da un'ora all'interno di un ciclo che finisce quando l'iteratore raggiunge 'X'.

BIBLIOTECHE_MENO_UTILIZZATE


```

CREATE PROCEDURE BibliotecheMenoUtilizzate()
BEGIN
CREATE VIEW PT AS
    SELECT NomeBiblioteca, COUNT(*) AS PostiTotali FROM POSTOLETTURA
    GROUP BY NomeBiblioteca;
CREATE VIEW OCC AS
    SELECT NomeBibliotecaPostoLettura, COUNT(*) AS PostiPrenotati FROM PRENOTAZIONEPOSTO
    GROUP BY NomeBibliotecaPostoLettura;
SELECT NomeBiblioteca
    FROM PT JOIN OCC ON NomeBiblioteca=NomeBibliotecaPostoLettura
    ORDER BY PostiTotali/PostiPrenotati DESC;
END //

```

La stored procedure BibliotecheMenoUtilizzate ha lo scopo di restituire una classifica con le biblioteche che sono meno utilizzate in proporzione alla loro grandezza.

Il rapporto tra il numero totale di posti lettura e il numero di prenotazioni è scelto proprio per rappresentare il concetto di biblioteca meno utilizzata specificato prima.

La procedure si basa su due view: OCC che seleziona il numero di prenotazioni di posti lettura nelle singole biblioteche; PT che seleziona il numero totale di posti lettura di ogni biblioteca. Infine, viene fatta l'operazione di JOIN tra le due view e si ordina decrescentemente il risultato per il rapporto tra i due valori delle view, ottenendo così le biblioteche dalla meno utilizzata.

ELIMINA_PRENOTAZIONE

```

CREATE PROCEDURE EliminaPrenotazioneLibro(IN EmailU VARCHAR(50), IN C INT)
BEGIN
    DECLARE T VARCHAR(50);
    DECLARE G INT;
    SET T = (SELECT EmailUtilizzatore
              FROM PRENOTAZIONELIBRO
              WHERE Codice=C);
    SET G = (SELECT CARTACEO.Codice
              FROM CARTACEO JOIN PRENOTAZIONELIBRO ON CARTACEO.Codice=PRENOTAZIONELIBRO.CodiceCartaceo
              WHERE PRENOTAZIONELIBRO.Codice=C);
    IF (T=EmailU) THEN
        DELETE FROM EVENTOCONSEGNA WHERE CodicePrenotazioneLibro=C;
        DELETE FROM PRENOTAZIONELIBRO WHERE Codice=C;
        UPDATE CARTACEO SET StatoPrestito='Disponibile' WHERE Codice=G;
    END IF;
END $

```

L'eliminazione di una riga in una tabella può essere problematica principalmente per due motivi:

- Nella riga ci potrebbero essere delle chiavi secondarie di altre tabelle;
- Un utente potrebbe sbagliarsi e provare a cancellare la prenotazione di un altro utente.

Per il primo problema la stored esplicitamente cancella anche le righe di altre tabelle che dipendono dalla riga da eliminare, come nello screen mostrato sopra dove, per togliere una riga di PRENOTAZIONELIBRO, viene anche eliminato una riga di EVENTOCONSEGNA e viene effettuato un UPDATE alla tabella cartaceo.

Per quanto riguarda il secondo, in tutte le stored utilizzano il comando 'DELETE' c'è sempre il controllo che la riga da eliminare appartenga alla mail della persona che sta cercando di lanciare l'operazione.

Nella stored presa in esame la variabile T seleziona l'email della riga che si vuole cancellare e qualora questa mail fosse uguale a quella data in input, ovvero quella dell'utente che vuole eliminare la riga, si procede alle operazioni di writing.

LIBRO_DEL_MESE

```
CREATE PROCEDURE LibroDelMese()
BEGIN
    SELECT DISTINCT CARTACEO.Titolo, COUNT(*) as NrPrenotazioni
    FROM CARTACEO JOIN PRENOTAZIONELIBRO ON (CARTACEO.Codice = PRENOTAZIONELIBRO.CodiceCartaceo)
    WHERE CURDATE()<=DataFine
    GROUP BY CARTACEO.Titolo
    ORDER BY NrPrenotazioni DESC
    LIMIT 10;
END $
```

Nella sezione statistiche può essere utile conoscere i trend dell'ultimo mese oltre alle classifiche che analizzano i dati dalla nascita della piattaforma.

Alcune stored procedure, quindi, semplicemente mettono un limite di tempo per i dati da ordinare con il semplice controllo che la data sia minore alla data del giorno in cui viene lanciata meno un mese.

POSTI_PRENOTATI_ATTIVI

```
CREATE PROCEDURE PostiPrenotatiAttivi(IN EmailU VARCHAR(50))
BEGIN
    SELECT NomeBibliotecaPostoLettura, NumeroPostoLettura, DataPrenotazione, OraInizio, OraFine
    FROM PRENOTAZIONEPOSTO
    WHERE (EmailUtilizzatore=EmailU) AND (CURDATE()<=DataPrenotazione);
END $
```

Anche nella visualizzazione delle azioni compiute nella piattaforma può essere importante avere un modo per vedere solo quelle che sono ancora attive.

Ad esempio, solo le prenotazioni di posti lettura che sono nel futuro e non anche quelle dei giorni scorsi. PostiPrenotatiAttivi restituisce solo le determinate prenotazioni che rispettano la condizione di avere la DataPrenotazione maggiore della data di oggi (CURDATE()). In questo modo è immediato per gli utenti poter vedere quali consegne dovranno fare o stanno aspettando senza che si mischino con tutte le operazioni del passato.

INVIA_SEGNALAZIONE

```
CREATE PROCEDURE InviaSegnalazione(IN EmailU VARCHAR(100), IN Testo1 VARCHAR(200), IN EmailAmm VARCHAR(100))
BEGIN
    DECLARE str VARCHAR(100);
    DECLARE x INT;

    SET x = (SELECT COUNT(*) FROM SEGNALAZIONE WHERE EmailUtilizzatore = EmailU);
    IF (x<3) THEN
        SET x = x + 1;
        INSERT INTO SEGNALAZIONE(EmailUtilizzatore, NrSegnalazione, DataSegnalazione, Testo, EmailAmministratore)
        VALUES (EmailU, x, CURDATE(), Testo1, EmailAmm);
    END IF;
END |
```

InviaSegnalazione permette ad un amministratore di inviare una segnalazione ad un utilizzatore.

Per evitare che l'amministratore debba mettere manualmente il numero di segnalazioni ricevute da un utilizzatore, il sistema gestisce autonomamente il numero di segnalazioni. Infatti, l'amministratore inserisce solo la mail del destinatario e il testo della segnalazione (la email dell'amministratore è una variabile di sessione che anch'essa viene mandata automaticamente) ed è la stored procedure che conta nel database il numero di segnalazioni che l'utilizzatore ha già ricevuto. Se il numero è minore di tre, la segnalazione verrà inserita con il numero trovato aggiungendo uno, se è uguale a tre allora verrà gestita da un trigger.

Il numero non potrà essere maggiore di tre in quanto alla terza segnalazione l'utente viene sospeso e non può più usare il portale Ebiblio.

TRIGGER

CATALOGA_LIBRO

```
CREATE TRIGGER LibroPrenotato
AFTER INSERT ON PRENOTAZIONELIBRO
FOR EACH ROW
UPDATE CARTACEO SET StatoPrestito = 'Prenotato' WHERE (NEW.CodiceCartaceo=CARTACEO.Codice);
```

Appena viene aggiunta una prenotazione di un libro cartaceo nella tabella PRENOTAZIONELIBRO, si attiva un trigger che cambia lo stato prestito del libro prenotato da 'Disponibile' a 'Prenotato'. In questo modo non è poi possibile prenotare quel libro fino a che l'attributo 'StatoPrestito' non torna 'Disponibile'.

INVIA_SEGNALAZIONE

```
CREATE TRIGGER TriggerEventoConsegna
AFTER INSERT ON EVENTOCONSEGNA
FOR EACH ROW BEGIN
    DECLARE C INT;
    SET C = (SELECT DISTINCT PRENOTAZIONELIBRO.CodiceCartaceo
             FROM PRENOTAZIONELIBRO JOIN EVENTOCONSEGNA ON PRENOTAZIONELIBRO.Codice = EVENTOCONSEGNA.CodicePrenotazioneLibro
             WHERE NEW.CodicePrenotazioneLibro = PRENOTAZIONELIBRO.Codice);
    IF (NEW.Tipo = 'Affidamento') THEN
        UPDATE CARTACEO SET CARTACEO.StatoPrestito = 'Consegnato' WHERE C = CARTACEO.Codice;
    ELSE
        UPDATE CARTACEO SET CARTACEO.StatoPrestito = 'Disponibile' WHERE C = CARTACEO.Codice;
    END IF;
```

Quando un utente riceve tre segnalazioni non può più usare il portale Ebiblio e lo stato del suo account deve essere impostato da 'Attivo' in 'Sospeso'. Un account sospeso poi non potrà più accedere e quindi utilizzare i servizi del portale.

Il trigger CambioStato controlla il numero dell'ultima segnalazione aggiunta alla tabella SEGNALAZIONE e se fosse un tre, interviene sull'attributo 'StatoAccount'.

NUOVO_EVENTO

```
CREATE TRIGGER CambioStato AFTER INSERT ON SEGNALAZIONE
FOR EACH ROW
BEGIN
    IF NEW.NrSegnalazione = 3 THEN
        UPDATE UTILIZZATORE SET StatoAccount = 'Sospeso' WHERE Email = NEW.EmailUtilizzatore;
    END IF;
END;
```

Gli eventi di consegna possono essere di due tipologie: 'Affidamento' e 'Restituzione'.

La prima rappresenta la consegna di un libro con uno stato prestito di 'Prenotato' all'utilizzatore che ha effettuato la prenotazione, mentre la seconda rappresenta il libro restituito dall'utilizzatore che viene riportato alla biblioteca da un volontario.

Se la riga inserita nella tabella EVENTOCONSEGNA è del primo tipo, il trigger TriggerEventoConsegna cambia l'attributo 'StatoPrestito' del cartaceo interessato dalla consegna in 'Consegnato'. Se la consegna inserita è dell'altra tipologia, ovvero 'Restituzione' allora il cartaceo sarà modificato in 'Disponibile' e sarà possibile riprenotare il libro di nuovo.

VISUALIZZA_EBOOK

```
CREATE TRIGGER LetturaEbook
AFTER INSERT ON LETTURA
FOR EACH ROW
UPDATE EBOOK SET NrAccessi = NrAccessi+1 WHERE (NEW.CodiceEbook=EBOOK.Codice)
```

Le letture degli ebook sono controllate dall'attributo 'NrAccessi' che viene aggiornato, aumentandone il valore di uno, quando viene inserita una riga nella tabella LETTURA. Infatti, nel momento in cui un utente visualizza uno specifico ebook ed accede al suo link, viene fatta un'operazione di 'INSERT' alla tabella LETTURA con il nome dell'utilizzatore e l'ebook visualizzato.