

Progetto SchooLib

Laboratorio Interdisciplinare B



Sviluppato da:

Luca Marzella, Mattia Tittoto, Alessandro Tognetti

Indice generale

Scelte algoritmiche.....	2
Protocollo Di Comunicazione.....	4
Descrizione del protocollo:.....	4
Elenco dei comandi disponibili:.....	4
Strumenti di sviluppo ed installazione.....	6
Diagrammi UML.....	7
Use-Case Diagram.....	7
Class diagram.....	7
Communication Diagram.....	8
Database.....	9
Ristrutturazione Delle Specifiche.....	9
Diagramma ER.....	9
Vincoli di integrità.....	9
Ristrutturazione Dello Schema ER.....	10
Schema Logico.....	10
Query Creazione Tabelle.....	11
Manuale Di Installazione.....	14
Manuale Utente.....	15
AppReader:.....	15
Schermata Di Login.....	15
Funzionalità.....	16
AppLibrarian:.....	17
Schermata Di Login.....	17
Funzionalità.....	17
Organizzazione Del Processo Di Sviluppo.....	20

Scelte algoritmiche

La comunicazione client server è gestita tramite il design pattern proxy, nello specifico usa una classe Proxy per gestire la comunicazione da client a server, e una classe Skeleton per gestire la comunicazione da server a client. Entrambe le classi implementano un'interfaccia (ServerServices) contenente il prototipo dei metodi di cui Proxy e Skeleton eseguono override.

I comandi del protocollo di comunicazione sono contenuti in un tipo enum. Il protocollo è in grado di rappresentare i servizi che il server offre ai client.

Il multicient è garantito dal server, che, ad ogni richiesta di connessione, passerà i riferimenti del socket associato alla richiesta, ad un'istanza di tipo Thread, gestita dalla classe Skeleton, in grado di offrire i vari servizi al client in oggetto.

I metodi, lato server, che gestiscono la connessione con la base di dati sono tutti sincronizzati, per garantire accesso contemporaneo e sicuro al database. Tali metodi sono presenti nella classe ServerServicesDBBridge.

L'accesso e la modifica dei dati in modo concorrente nella base di dati viene garantita direttamente dal DBMS di Postgre.

La classe Utility contiene metodi statici che offrono servizi per la manipolazione dei dati, per esempio per eseguire controlli sulle stringhe. Questi metodi sono utilizzati da diverse classi presenti nel progetto.

Il codice usato per gestire l'email è un adattamento del codice scaricato da elearning, presente nella classe EmailSender.

Durante lo sviluppo software (stesura del codice) si è cercato il più possibile di separare la parte grafica dell'applicazione, dalla logica di controllo e dal contenuto informativo dei dati, cercando di adottare il pattern MVC (model view controller). Le classi relative la parte grafica sono presenti all'interno del package ClientSchoolLib (nel quale sono presenti tutte le classi comuni a Reader e Librarian), essendo condivise fra Reader e Librarian. La logica applicativa è implementata in appReader e appLibrarian. Le classi al loro interno estenderanno le sopracitate vesti grafiche da ClientSchoolLib (Es: AuthenticationLibrarian estende Authentication). Le eventuali differenze di una stessa interfaccia, sono opportunamente gestite da un controllo.

Si accede ai componenti grafici tramite metodi setter e getter.

Vi è poi un thread gestito dalla classe BookingDateChecket, lanciato dal main del server, che è quasi sempre in sleep e che fa un controllo periodico delle prenotazioni, per vedere chi ha attivo un prestito da più di una settimana. Questo controllo viene effettuato ogni 6 ore, per un totale di 4 volte al giorno.

Una classe gestisce l'inizializzazione e backup del database (DatabaseCreator). Opera su un file .csv dal quale preleva le query per la creazione delle tabelle e i dati di default. E' possibile anche usare dei dati di backup da un percorso indicato dall'utente all'avvio del programma. Questa classe offre anche la possibilità di fare un backup del server.

Sono presenti classi per la modellazione di prestiti, libri e prenotazioni, che offrono nella loro interfaccia la possibilità di gestire le informazioni dei sopracitati oggetti (Es: Book, Loan, Booking ecc.). Sono state realizzate per agevolare l'invio dei dati fra client e server, in modo da inviare tutti i dati insieme ed accedervi tramite metodi getter.

Di fatto sono classi serializzabili, con tanto di campo statico serialVersionUID.

Protocollo Di Comunicazione

Descrizione del protocollo:

il protocollo di comunicazione tra client e server è definito secondo il seguente schema:

<Tipo di comando>: è il servizio richiesto al server. I comandi disponibili sono definiti nel tipo enumerativo "ProtocolCmd".

<Parametri del comando>: è l'elenco dei parametri che il client invia al server necessari al server per eseguire il servizio richiesto.

<Valore di ritorno dal server>: è il valore che il server restituisce al client dopo aver eseguito il servizio richiesto.

Elenco dei comandi disponibili:

Nella tabella seguente sono indicati tutti i comandi disponibili (prima colonna). Nella seconda colonna vi è una breve descrizione del tipo di comando. Nella terza colonna sono indicati i parametri di ciascun comando. Per ogni parametro è indicato il tipo e l'eventuale posizione all'interno dell'array se il tipo è un array. Nell'ultima colonna invece è indicata la tipologia del valore di ritorno, con eventuali possibili valori.

Tipo di comando	Descrizione	Parametri del comando	Valore di ritorno dal server
ADD_BOOK	Richiesta aggiunta di un nuovo libro.	BookData: dati libro.	String: Esito operazione
AUTHENTICATION	Richiesta autenticazione di un utente.	String []: [0] id utente; [1] password utente; [2] codice autenticazione.	String: "autenticazione ok"; "utente inesistente"; "password errata"; "codice autenticazione errato".
BOOKING_BOOK	Richiesta prenotazione libro.	String []: [0] id utente; [1] codice libro; LocalDateTime: data prenotazione.	String: Esito operazione
CLOSE_BOOKING	Richiesta chiusura prenotazione.	String []: [0] id utente; [1] codice libro; LocalDateTime: data chiusura.	String: Esito operazione
CLOSE_CONNECTION	Richiesta chiusura connessione tra client e server.	Nessuno.	Nessuno.
CLOSE_LOAN	Richiesta chiusura prestito.	String []: [0] id utente; [1] codice libro; LocalDateTime: data chiusura.	String: Esito operazione
DELETE_BOOKING	Richiesta cancellazione prenotazione.	String []: [0] id utente; [1] codice libro.	String: Esito operazione
DELETE_USER	Richiesta cancellazione di un utente.	String []: [0] id utente; [1] password utente.	String: Esito operazione
GET_ALL_BOOKING	Richiesta elenco libri prenotati.	Nessuno.	Booking []: elenco libri prenotati.
GET_BOOKING_BY_BOOK	Richiesta prenotazioni associate ad un libro.	String: codice libro.	Booking []: elenco prenotazioni.
GET_BOOKING_BY_USER	Richiesta prenotazioni associate ad un utente.	String: id utente.	Booking []: elenco prenotazioni.
GET_BOOKS_BY_AUTHOR	Richiesta elenco dei libri per autore.	String []: [0] nome autore; [1] cognome autore.	BookData []: elenco libri.

GET_BOOKS_BY_CATEGORY	Richiesta elenco dei libri per categoria.	String: categoria	BookData []: elenco libri.
GET_BOOKS_BY_TITLE	Richiesta elenco dei libri per parola contenuta nel titolo.	String: parola da cercare.	BookData []: elenco libri.
GET_HISTORY_LOAN_BY_USER	Richiesta elenco prestiti effettuati da un utente.	String: id utente.	Loan []: elenco prestiti.
GET_MOST_READ_BOOK	Richiesta classifica libri più letti.	Nessuno.	BookData []: elenco libri più letti. All'indice 0 il libro più letto e a seguire gli altri in ordine decrescente.

Strumenti di sviluppo ed installazione

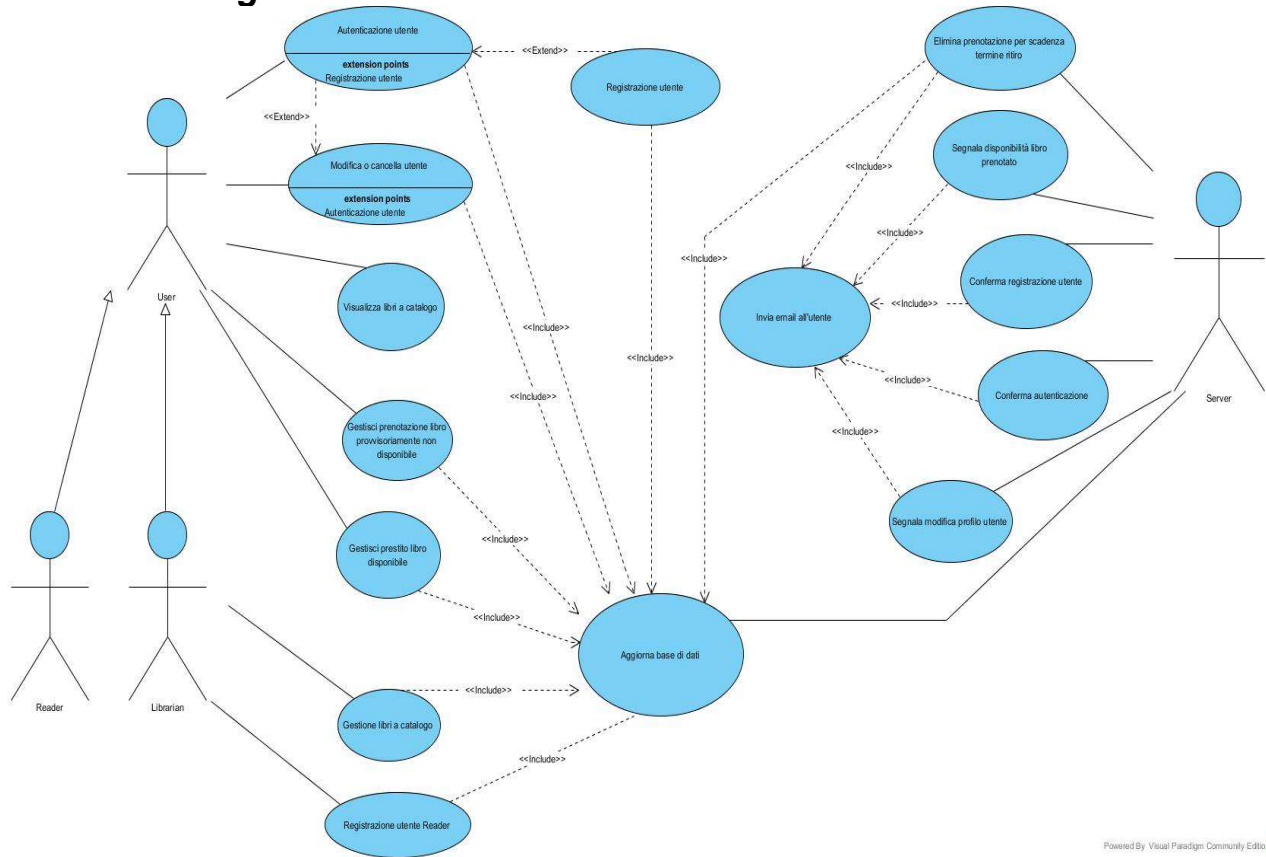
La versione utilizzata per la realizzazione del codice è NetBeans IDE 8.2, il testing delle varie funzionalità è stato anch'esso eseguito su questo software.

Per il database è stato utilizzato PgAdmin III PostgreSQL Tools v1.22.1. Le query sono state tutte provate più volte sul software sopra citato per accertarci delle loro funzionalità prima di inserirle all'interno del programma. Nota: essendo le query testate su questo software, l'utilizzo di un altro software potrebbe non leggere correttamente le suddette e, di conseguenza, non far funzionare il programma, si richiede quindi l'installazione di PgAdmin III prima di avviare il server.

Per la compilazione e l'esecuzione è stato utilizzato Apache Ant v1.10.1. I file build.xml sono stati customizzati per le 3 applicazioni da eseguire (SerSchoolLib, AppReader e AppLibrarian) in modo da permettere una compilazione con conseguente esecuzione automatica tramite Ant.

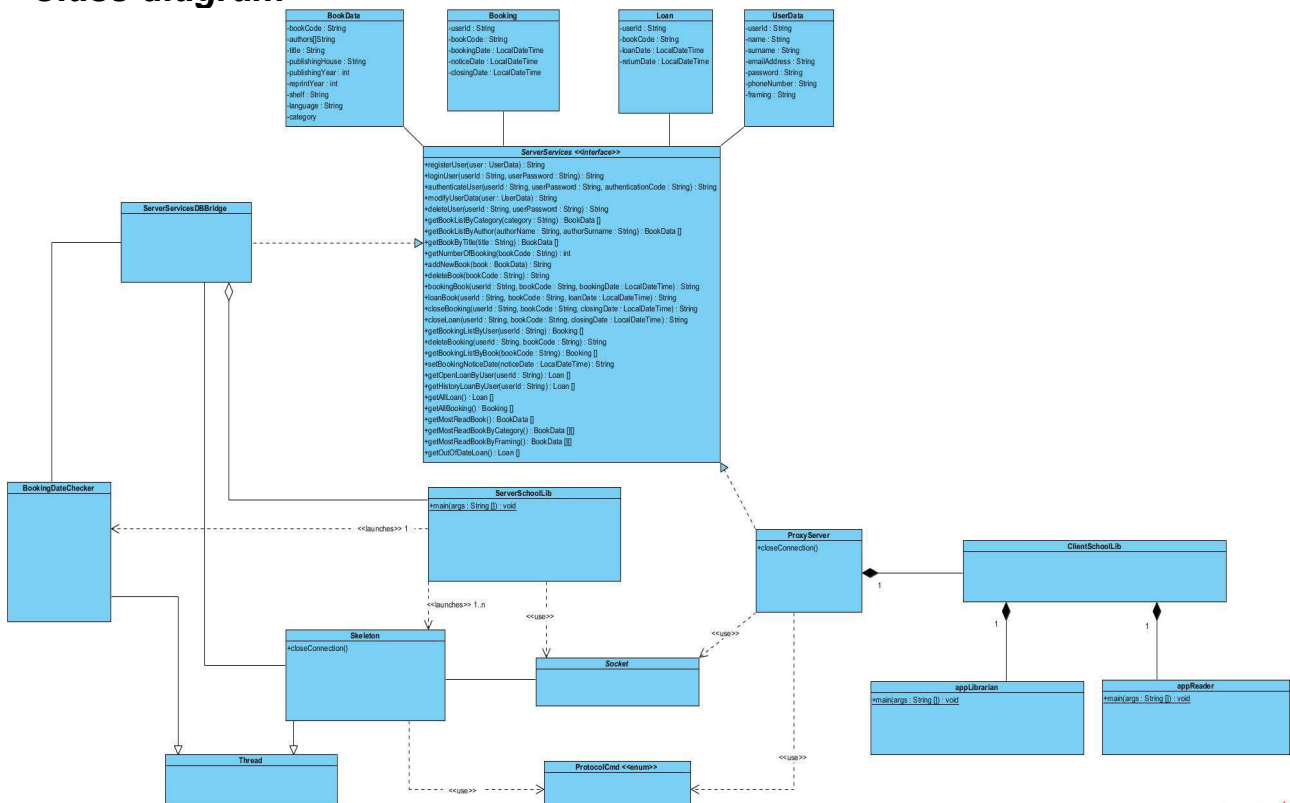
Diagrammi UML

Use-Case Diagram



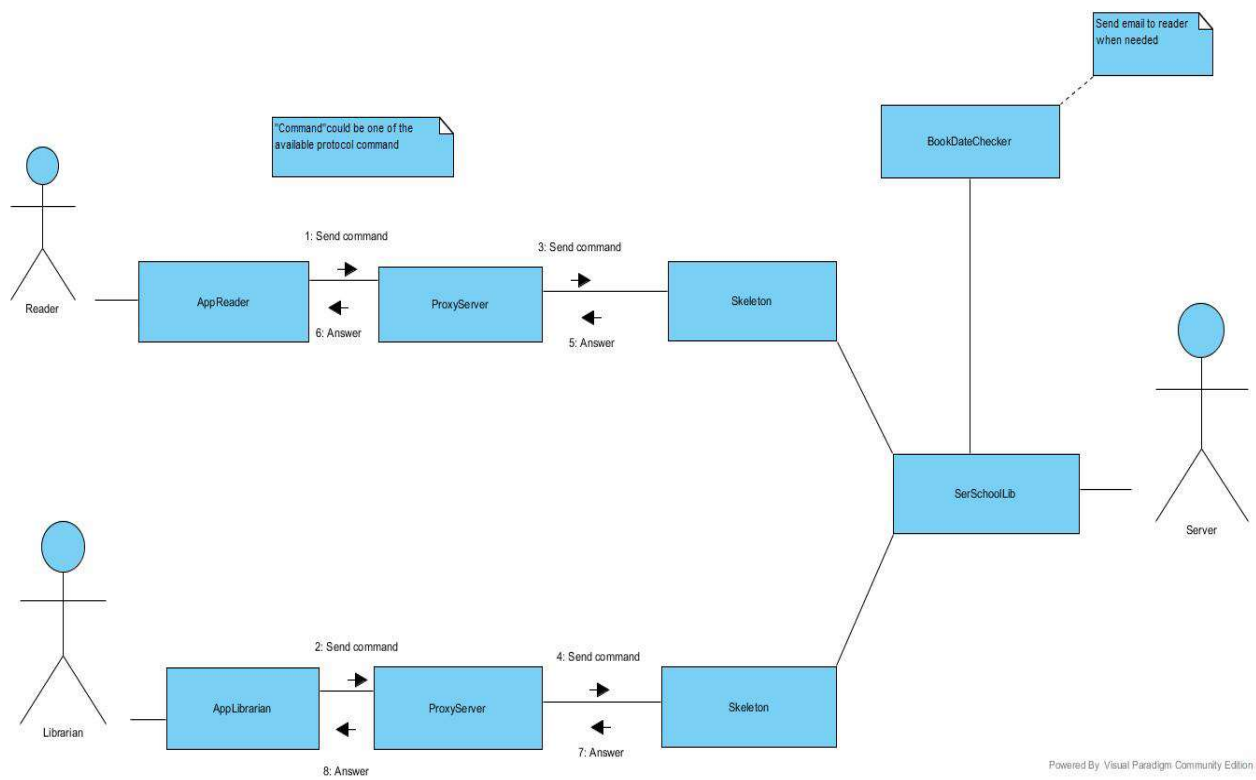
Powered By Visual Paradigm Community Edition

Class diagram



Powered By Visual Paradigm Community Edition

Communication Diagram

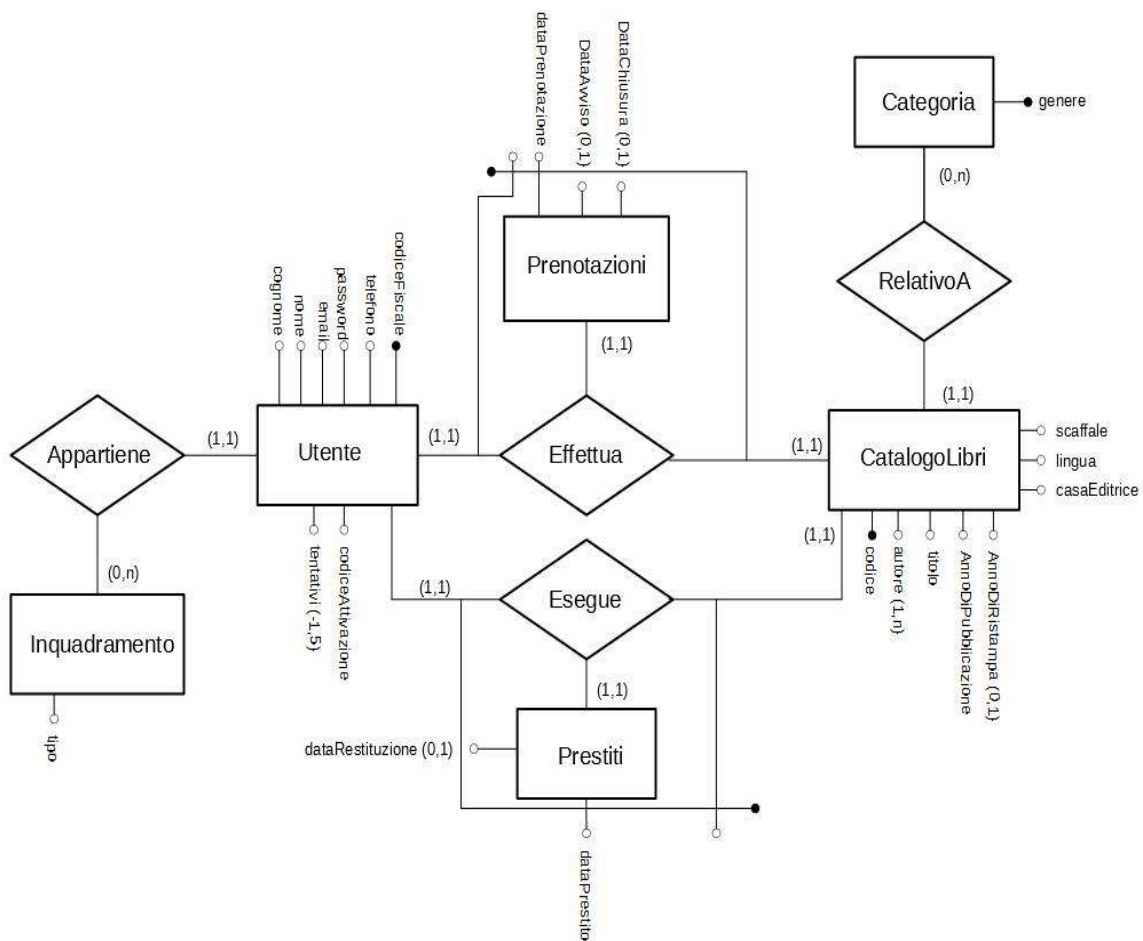


Database

Ristrutturazione Delle Specifiche

Non è stato ritenuto necessario ristrutturare le specifiche fornite per lo sviluppo del progetto.

Diagramma ER



Vincoli di integrità

- Utente può avere aperte al massimo 10 prenotazioni;
- Utente può avere in prestito al massimo 5 libri;
- Il campo tentativi di Utente può assumere valore intero nell'intervallo [-1,5].

Ristrutturazione Dello Schema ER



Schema Logico

- Utente(codiceFiscale, nome, cognome, email, password, telefono, codiceAttivazione, numeroTentativi, ^{Inquadramento}inquadramento)
- Inquadramento(tipo)
- CatalogoLibri(codice, titolo, casaEditrice, annoDiPubblicazione, annoDiRistampa_o, scaffale, lingua, genere^{Categoria})
- Autori(nome, cognome)
- ScrittoDa(codice^{CatalogoLibri}, nome^{Autori}, cognome^{Autori})
- Prestiti(codiceUtente^{Utenti}, codiceLibro^{CatalogoLibri}, dataPrestito, dataRestituzione)
- Prenotazioni(codiceUtente^{Utenti}, codiceLibro^{CatalogoLibri}, dataPrenotazione, dataChiusura)
- Categoria (genere)

Query Creazione Tabelle

INQUADRAMENTO

```
create table inquadramento
(  
    tipo varchar(100) primary key  
);
```

UTENTE

```
create table utente
(  
    CF varchar(16) primary key,  
    nome varchar(30) NOT NULL,  
    cognome varchar(30) NOT NULL,  
    email varchar(50) NOT NULL,  
    pw varchar(12) NOT NULL,  
    telefono integer NOT NULL,  
    codiceAttivazione varchar(5) NOT NULL,  
    nTentativi integer CHECK(nTentativi between -1 and 5),  
    autenticato boolean NOT NULL,  
    inquadramento varchar(100) references inquadramento  
    on delete set default  
    on update cascade  
);
```

CATEGORIA

```
create table categoria
(  
    genere varchar(25) primary key  
);
```

CATALOGO LIBRI

```
create table catalogoLibri
(
  codice varchar(10) primary key,
  titolo varchar(30) NOT NULL,
  casaEditrice varchar(30) NOT NULL,
  annoDiPubblicazione date NOT NULL,
  annoDiRistampa date,
  scaffale varchar(3) NOT NULL,
  lingua varchar(10) NOT NULL,
  genere varchar(25) references categoria
  on delete no action
  on update cascade
);
```

AUTORI

```
create table autore
(
  nomeAutore varchar(20),
  cognomeAutore varchar(20),
  primary key(nomeAutore,cognomeAutore)
);
```

SCRITTODA

```
create table scrittoDa
(
  codice varchar(10),
  nomeAutore varchar(20),
  cognomeAutore varchar(20),
  primary key(codice,nomeAutore,cognomeAutore),
  foreign key(codice) references catalogolibri
  on delete cascade
  on update cascade,
  foreign key(nomeAutore,cognomeAutore) references autore
  on delete cascade
  on update cascade
);
```

PRESTITO

create table prestito

```
(  
  codiceUtente varchar(16),  
  codiceLibro varchar(10),  
  dataPrestito date,  
  dataRestituzione date,  
  primary key(codiceUtente,codiceLibro),  
  foreign key (codiceUtente) references utente  
  on delete cascade  
  on update cascade,  
  foreign key (codiceLibro) references catalogolibri  
  on delete cascade  
  on update cascade  
);
```

PRENOTAZIONE

create table prenotazione

```
(  
  codiceUtente varchar(16),  
  codiceLibro varchar(10),  
  dataPrenotazione date,  
  dataChiusura date,  
  primary key(codiceUtente,codiceLibro,dataPrenotazione),  
  foreign key (codiceUtente) references utente  
  on delete cascade  
  on update cascade,  
  foreign key (codiceLibro) references catalogolibri  
  on delete cascade  
  on update cascade  
);
```

Manuale Di Installazione

Prima di tutto, disattivare l'antivirus sulla macchina sulla quale verrà eseguita l'applicazione server, o l'invio dell'email non funzionerà (testato con avast free antivirus).

Queste applicazioni richiedono di aver installati e funzionanti i seguenti programmi:

- JDK 1.6 o successivo
- PGAdmin 3
- Apache ant con relative variabili d'ambiente correttamente impostate

Verificare il corretto funzionamento di questi programmi e poi proseguire con l'esecuzione delle applicazioni.

Passo 1: compilare ed eseguire il server in caso non sia mai stato compilato

- Aprire una finestra di comando nella directory ".\serSchoolLib"
- Eseguire ant.bat digitando "ant" (il file build.xml dovrebbe venire automaticamente riconosciuto)
- Seguire le istruzioni a schermo per interagire con il server

Passo 1.2: eseguire il server in caso sia già stato compilato in precedenza

- Aprire una finestra di comando nella directory ".\serSchoolLib"
- Eseguire ant.bat digitando "ant runServer" (il file build.xml dovrebbe venire automaticamente riconosciuto)
- Seguire le istruzioni a schermo per interagire con il server

Passo 2: compilare ed eseguire il client per l'utente Reader o per l'utente Librarian in caso non sia mai stato compilato

- **Per l'utente Reader**
 - Aprire una finestra di comando nella directory ".\appReader"
 - Eseguire ant.bat digitando "ant" (il file build.xml dovrebbe venire automaticamente riconosciuto)
 - Seguire le istruzioni a schermo per interagire con il client
- **Per l'utente Librarian**
 - Aprire una finestra di comando nella directory ".\appLibrarian"
 - Eseguire ant.bat digitando "ant" (il file build.xml dovrebbe venire automaticamente riconosciuto)
 - Seguire le istruzioni a schermo per interagire con il client

Passo 2.2: eseguire il client per l'utente Reader o per l'utente Librarian in caso sia già stato compilato in precedenza

- **Per l'utente Reader**

- Aprire una finestra di comando nella directory ".\appReader"
- Eseguire ant.bat digitando "ant runReader" (il file build.xml dovrebbe venire automaticamente riconosciuto)
- Seguire le istruzioni a schermo per interagire con il client

- **Per l'utente Librarian**

- Aprire una finestra di comando nella directory ".\appLibrarian"
- Eseguire ant.bat digitando "ant runLibrarian" (il file build.xml dovrebbe venire automaticamente riconosciuto)
- Seguire le istruzioni a schermo per interagire con il client

Manuale Utente

Appena aperta l'applicazione, inserire la porta di comunicazione del server (default: 8080) e premere conferma.

AppReader:

Schermata Di Login

Login Utente: Inserire i dati (Lo username è il Codice fiscale inserito nel momento della registrazione dell'utente) e premere successivamente "Login" per effettuare il login di un utente già esistente.

Registrazione Nuovo Utente: Premere "Registrazione" e compilare i campi richiesti nella schermata di registrazione.

Recupero Password Utente: Premere "Password Dimenticata" e inserire lo username nella schermata di recupero password.

In caso di utente non attivo, dopo il login comparirà una schermata per l'attivazione dell'utente, al che inserire il codice d'attivazione ricevuto per email al momento della registrazione e premere "Conferma".

Funzionalità

Gestione Profilo: tramite questo bottone è possibile modificare i dati relativi al proprio profilo ed, eventualmente, cancellarlo. Verrà aperta una schermata di gestione del profilo.

- **Modifica Password:** permette di modificare la password dell'utente. Viene aperta una schermata in cui inserire la nuova password.
- **Modifica Email:** permette di modificare l'email dell'utente. Viene aperta una schermata in cui inserire la nuova email.
- **Modifica Numero Di Telefono:** permette di modificare il numero di telefono dell'utente. Viene aperta una schermata in cui inserire il nuovo numero di telefono.
- **Modifica Inquadramento:** permette di modificare l'inquadramento dell'utente. Viene aperta una schermata in cui selezionare il nuovo inquadramento.
- **Cancellazione Profilo:** permette di cancellare il profilo. Viene aperta una schermata in cui inserire l'attuale password per confermare la cancellazione del profilo.

Gestione Catalogo: tramite questo bottone è possibile effettuare ricerche, secondo diversi criteri, dei libri presenti nel catalogo. Verrà aperta una schermata di gestione del catalogo.

- **Ricerca Per Categoria:** permette di ricercare dei libri appartenenti ad una categoria. Viene aperta una schermata in cui selezionare la categoria interessata.
- **Ricerca Per Autore:** permette di ricercare dei libri scritti da un autore specifico. Viene aperta una schermata in cui inserire il nome dell'autore interessato.
- **Ricerca Per Titolo:** permette di ricercare dei libri con un determinato titolo. Viene aperta una schermata in cui inserire il titolo interessato

I risultati della ricerca verranno mostrati in una pagina a parte fino ad un massimo di 25 libri alla volta. E' possibile mostrare altri libri (se presenti) premendo i tasti "Prossimi" e "Precedenti" presenti nella schermata dei risultati della ricerca.

Gestione Prenotazioni: tramite questo bottone è possibile prenotare un libro, cancellare una prenotazione in corso per l'utente e visualizzare le proprie prenotazioni in corso.

- **Prenota:** permette di prenotare un libro. Viene aperta una schermata in cui inserire il codice del libro da prenotare.
- **Cancella Prenotazione** : permette di cancellare la prenotazione di un libro. Viene aperta una schermata in cui inserire il codice del libro per cui cancellare la prenotazione.
- **Prenotazioni In Corso:** viene aperta una schermata contenente le prenotazioni in corso dell'utente.

Esci: effettua il logout dell'utente.

AppLibrarian:

Schermata Di Login

Login Utente: Inserire i dati (Lo username è il Codice fiscale inserito nel momento della registrazione dell'utente) e premere successivamente “Login” per effettuare il login di un utente già esistente.

Registrazione Nuovo Utente: Premere “Registrazione” e compilare i campi richiesti nella schermata di registrazione.

Recupero Password Utente: Premere “Password Dimenticata” e inserire lo username nella schermata di recupero password.

In caso di utente non attivo, dopo il login comparirà una schermata per l'attivazione dell'utente, al che inserire il codice d'attivazione ricevuto per email al momento della registrazione e premere “Conferma”.

Funzionalità

Gestione Profilo: tramite questo bottone è possibile modificare i dati relativi al proprio profilo ed, eventualmente, cancellarlo. Verrà aperta una schermata di gestione del profilo.

- **Modifica Password:** permette di modificare la password dell'utente. Viene aperta una schermata in cui inserire la nuova password.
- **Modifica Email:** permette di modificare l'email dell'utente. Viene aperta una schermata in cui inserire la nuova email.
- **Modifica Numero Di Telefono:** permette di modificare il numero di telefono dell'utente. Viene aperta una schermata in cui inserire il nuovo numero di telefono.
- **Modifica Inquadramento:** permette di modificare l'inquadramento dell'utente. Viene aperta una schermata in cui selezionare il nuovo inquadramento.
- **Cancellazione Profilo:** permette di cancellare il profilo. Viene aperta una schermata in cui inserire l'attuale password per confermare la cancellazione del profilo.

Gestione Catalogo: tramite questo bottone è possibile effettuare ricerche, secondo diversi criteri, dei libri presenti nel catalogo, inserire libri nel catalogo o eliminare libri dal catalogo. Verrà aperta una schermata di gestione del catalogo.

- **Ricerca Per Categoria:** permette di ricercare dei libri appartenenti ad una categoria. Viene aperta una schermata in cui selezionare la categoria interessata.
- **Ricerca Per Autore:** permette di ricercare dei libri scritti da un autore specifico. Viene aperta una schermata in cui inserire il nome dell'autore interessato.
- **Ricerca Per Titolo:** permette di ricercare dei libri con un determinato titolo. Viene aperta una schermata in cui inserire il titolo interessato

I risultati delle ricerche verranno mostrati in una pagina a parte fino ad un massimo di 25 libri alla volta. E' possibile mostrare altri libri (se presenti) premendo i tasti "Prossimi" e "Precedenti" presenti nella schermata dei risultati della ricerca.

- **Inserisci Nuovo Libro:** permette di inserire nuovi libri nel catalogo. Viene aperta una schermata in cui inserire i dati relativi al libro da inserire in catalogo.
- **Elimina Libro:** permette di cancellare libri dal catalogo. Viene aperta una schermata in cui inserire il codice del libro da eliminare dal catalogo.

Gestione Prenotazioni: tramite questo bottone è possibile prenotare un libro, cancellare una prenotazione in corso per l'utente e visualizzare le proprie prenotazioni in corso.

- **Prenota:** permette di prenotare un libro. Viene aperta una schermata in cui inserire il codice del libro da prenotare.
- **Cancella Prenotazione:** permette di cancellare la prenotazione di un libro. Viene aperta una schermata in cui inserire il codice del libro per cui cancellare la prenotazione.
- **Prenotazioni In Corso:** viene aperta una schermata contenente le prenotazioni in corso dell'utente.
- **Prenota Libro Per Utente:** permette di prenotare un libro per conto di un utente. Viene aperta una schermata in cui inserire il codice del libro da prenotare e il codice utente per il quale prenotarlo.
- **Cancella Prenotazione Utente:** permette di cancellare la prenotazione di un libro. Viene aperta una schermata in cui inserire il codice del libro e il codice utente per cui cancellare la prenotazione.
- **Mostra Prenotazioni Utente:** viene aperta una schermata in cui inserire il codice utente del quale visualizzare le prenotazioni. Successivamente viene aperta una schermata contenente le prenotazioni in corso dell'utente inserito.

Gestione Utenti: tramite questo bottone è possibile registrare utenti al servizio o cancellarli.

- **Registra Utente:** permette di registrare un utente al servizio. Viene aperta una schermata in cui inserire i dati dell'utente da registrare nei relativi campi.
- **Cancella Utente:** permette di cancellare un utente dal servizio. Viene aperta una schermata in cui inserire il codice utente da cancellare.

Gestione Prestiti: tramite questo bottone è possibile registrare prestiti, cancellare prestiti o registrare la restituzione di un libro da parte di un utente.

- **Registra Un Prestito:** permette di registrare un prestito a nome di un utente. Viene aperta una schermata in cui inserire il codice utente che effettua il prestito e il codice del libro da dare in prestito al suddetto.
- **Cancellazione Prestito:** permette di cancellare un prestito in corso. Viene aperta una schermata in cui inserire il codice del libro prestato del quale cancellare il prestito.
- **Restituzione Prestito:** permette di registrare la restituzione di un libro. Viene aperta una schermata in cui inserire il codice del libro restituito.

Analisi Biblioteca: tramite questo bottone è possibile effettuare diverse azioni relative allo stato dell'intero sistema bibliotecario.

- **Mostra I Prestiti Di Tutti Gli Utenti:** viene aperta una schermata in cui vengono mostrati tutti i libri attualmente in prestito.
- **Mostra Prestiti Di Un Utente:** viene aperta una schermata in cui inserire il codice utente del quale vedere i prestiti. Successivamente, viene aperta una schermata in cui vengono mostrati tutti i prestiti del suddetto utente.
- **Mostra Le Prenotazioni Di Tutti Gli Utenti:** viene aperta una schermata in cui vengono mostrate tutte le prenotazioni degli utenti.
- **Mostra Lo Storico Di Un Utente:** viene aperta una schermata in cui inserire il codice utente del quale vedere lo storico. Successivamente, viene aperta una schermata in cui viene mostrato lo storico del suddetto utente.
- **Mostra I Più Letti In Assoluto:** viene aperta una schermata in cui vengono mostrati i libri più letti.
- **Mostra I Più Letti Per Categoria:** viene aperta una schermata in cui selezionare la categoria della quale vedere i libri più letti. Successivamente, viene aperta una schermata in cui vengono mostrati i libri più letti appartenenti a quella categoria.
- **Mostra I Più Letti Per Inquadramento:** viene aperta una schermata in cui selezionare l'inquadramento di un utente. Successivamente, viene aperta una schermata in cui vengono mostrati i libri più letti dagli utenti appartenenti a quell'inquadramento.

- **Mostra Prestiti Sforanti:** viene aperta una schermata in cui vengono mostrati i prestiti che hanno superato il tempo massimo di concessione del prestito.

Esci: effettua il logout dell'utente.

Organizzazione Del Processo Di Sviluppo

Fase del processo	Strumenti utilizzati	Esecutore della fase
Analisi dei requisiti		Alessandro Tognetti, Luca Marzella, Mattia Tittoto
Suddivisione e pianificazione delle attività, con relative date di scadenza		Alessandro Tognetti, Luca Marzella, Mattia Tittoto
Progettazione: sviluppo dei diagrammi UML	Visual paradigm	Mattia Tittoto
Progettazione della base di dati (ristrutturazione delle specifiche, diagramma ER, ristrutturazione diagramma ER, schema logico)	LibreOffice 5.3 (draw)	Mattia Tittoto
Definizione del protocollo di comunicazione tra client e server		Luca Marzella, Mattia Tittoto
Definizione delle query per tutte le interazioni con la base di dati		Luca Marzella
Codifica in linguaggio Java del prodotto SW	Netbeans 8.1, JDK 1.8.0_121	Luca Marzella
Rivisitazione del codice sorgente e inserimento dei commenti per la creazione della documentazione con javaDoc	Netbeans 8.1, JDK 1.8.0_121	Alessandro Tognetti
Debug del prodotto SW (fase iniziale di testing per catturare i macro bug, ovvero gli errori più facili da intercettare)	Netbeans 8.1, JDK 1.8.0_121	Luca Marzella
Debug del prodotto SW (fase finale di testing per catturare i micro bug, ovvero gli errori più difficili da intercettare)	Netbeans 8.1, JDK 1.8.0_121	Alessandro Tognetti
Compilazione per la creazione della documentazione del prodotto SW	javaDoc	Alessandro Tognetti
Sviluppo e compilazione del file di installazione	Ant	Alessandro Tognetti

Creazione del manuale utente e di installazione		Alessandro Tognetti
Integrazione del prodotto SW su una macchina sprovvista di tutti gli elementi che compongono il sistema finale		Luca Marzella
Creazione del file che contiene tutte le parti che documentano il progetto		Alessandro Tognetti, Mattia Tittoto
Rivisitazione del progetto completo		Alessandro Tognetti, Luca Marzella, Mattia Tittoto