**I.T.U.**

**Faculty of Computer and Informatics**

**Computer Engineering**



**DATABASE MANAGAMENT SYSTEMS**

**E-LIBRARY PROJECT REPORT**

# CONTENTS

[CONTENTS 1](#_Toc375493532)

[PART I 4](#_Toc375493533)

[1 Genarel Information 4](#_Toc375493534)

[1.1 Project Description 5](#_Toc375493535)

[1.2 Development and Execution Environment 5](#_Toc375493536)

[1.3 Tasks of Group Members 5](#_Toc375493537)

[1.3.1 Mustafa Uçar 5](#_Toc375493538)

[1.3.2 Tugrul Yatağan 5](#_Toc375493539)

[1.3.3 Emre Gökrem 5](#_Toc375493540)

[1.3.4 Hüseyin Erdoğan 6](#_Toc375493541)

[PART II 7](#_Toc375493542)

[2 Setup 7](#_Toc375493543)

[2.1 Database Creation 8](#_Toc375493544)

[2.2 Table Creation 8](#_Toc375493545)

[2.3 Connection to MySQL Server 8](#_Toc375493546)

[PART III 9](#_Toc375493547)

[3 User Manual 9](#_Toc375493548)

[3.1 Guest Interface 10](#_Toc375493549)

[3.1.1 Guest Home Page 10](#_Toc375493550)

[3.1.2 Guest Search Procedure 10](#_Toc375493551)

[3.1.3 Most Popular Sources 13](#_Toc375493552)

[3.1.4 Request & Suggestions Page 15](#_Toc375493553)

[3.1.5 Contact Us Page 15](#_Toc375493554)

[3.1.6 About Us Page 16](#_Toc375493555)

[3.2 User Interface 17](#_Toc375493556)

[3.2.1 Register 17](#_Toc375493557)

[3.2.2 Login 17](#_Toc375493558)

[3.2.3 Logout 18](#_Toc375493559)

[3.2.4 User Home Page 18](#_Toc375493560)

[3.2.5 User Searching Procedure 19](#_Toc375493561)

[3.2.6 Most Popular Page 21](#_Toc375493562)

[3.2.7 User Profile 21](#_Toc375493563)

[3.2.8 User Edit Profile 22](#_Toc375493564)

[3.2.9 User Archive 24](#_Toc375493565)

[3.3 Admin Interface 24](#_Toc375493566)

[3.3.1 Admin Home Page 24](#_Toc375493567)

[3.3.2 Admin Profile 24](#_Toc375493568)

[3.3.3 Admin Edit Profile 25](#_Toc375493569)

[3.3.4 Admin Search Procedure 26](#_Toc375493570)

[3.3.4.1 Source Placed Page 27](#_Toc375493571)

[3.3.4.2 Source’s Records 27](#_Toc375493572)

[3.3.4.3 Source’s Records Edit 28](#_Toc375493573)

[3.3.4.4 Source Edit 28](#_Toc375493574)

[3.3.4.5 Source Delete 30](#_Toc375493575)

[3.3.5 Adding Source 30](#_Toc375493576)

[3.3.5.1 Adding Book 31](#_Toc375493577)

[3.3.5.2 Adding DVD 32](#_Toc375493578)

[3.3.5.3 Adding Magazine 32](#_Toc375493579)

[3.3.6 Admin Archive 33](#_Toc375493580)

[PART IV 34](#_Toc375493581)

[4 Technical Manual 34](#_Toc375493582)

[4.1 Database Design 35](#_Toc375493583)

[4.1.1 Tables 35](#_Toc375493584)

[4.1.1.1 “users” Table 35](#_Toc375493585)

[4.1.1.2 “logs” Table 35](#_Toc375493586)

[4.1.1.3 “authors” Table 35](#_Toc375493587)

[4.1.1.4 “books” Table 36](#_Toc375493588)

[4.1.1.5 “magazines” Table 36](#_Toc375493589)

[4.1.1.6 “DVDs” Table 36](#_Toc375493590)

[4.1.1.7 “book\_categories” Table 37](#_Toc375493591)

[4.1.1.8 “bookcases” Table 37](#_Toc375493592)

[4.1.1.9 “bookshelves” Table 37](#_Toc375493593)

[4.1.1.10 “DVD\_categories” Table 38](#_Toc375493594)

[4.1.1.11 “electronic\_resources” Table 38](#_Toc375493595)

[4.1.1.12 “physical\_resources” Table 38](#_Toc375493596)

[4.1.1.13 “waiting\_resources” Table 38](#_Toc375493597)

[4.1.1.14 “floors” Table 39](#_Toc375493598)

[4.1.1.15 “libraries” Table 39](#_Toc375493599)

[4.1.1.16 “magazine\_categories” Table 39](#_Toc375493600)

[4.1.1.17 “records” Table 39](#_Toc375493601)

[4.1.1.18 “suggestions” Table 40](#_Toc375493602)

[4.1.1.19 “types” Table 40](#_Toc375493603)

[4.1.2 Views 40](#_Toc375493604)

[4.1.2.1 “book\_information” View 40](#_Toc375493605)

[4.1.2.2 “DVD\_information” View 41](#_Toc375493606)

[4.1.2.3 “electronic\_books” View 41](#_Toc375493607)

[4.1.2.4 “electronic\_magazines” View 41](#_Toc375493608)

[4.1.2.5 “magazine\_informations” View 41](#_Toc375493609)

[4.1.2.6 “physical\_books” View 42](#_Toc375493610)

[4.1.2.7 “physical\_magazines” View 42](#_Toc375493611)

[4.1.3 Foreign Keys 43](#_Toc375493612)

[4.1.3.1 In “bookscases” table 43](#_Toc375493613)

[4.1.3.2 In “bookshelves” table 43](#_Toc375493614)

[4.1.3.3 In “electronic\_resources” table 43](#_Toc375493615)

[4.1.3.4 In “floors” table 43](#_Toc375493616)

[4.1.3.5 In “physical\_resources” table 43](#_Toc375493617)

[4.1.4 Entity Relationship Diagram 43](#_Toc375493618)

[4.2 Software Design 44](#_Toc375493619)

[4.2.1 Admin Classes 44](#_Toc375493620)

[4.2.2 User Classes 44](#_Toc375493621)

[4.2.3 Book Classes 44](#_Toc375493622)

[4.2.4 Magazine Classes 44](#_Toc375493623)

[4.2.5 DVD Classes 44](#_Toc375493624)

[4.2.6 Additional Classes 44](#_Toc375493625)

# PART I

# Genarel Information

## Project Description

The aim of this project is providing whole library system which is accessible via web. Users will be able to see sources (book, e-book, DVD, magazine etc.) and if users are member of the system, they will be rent sources.

Users will be able to find the location of the sources from website and they can take it actual material from library. If a user wants to use electronic sources from website, he/she will be able to find information about electronic source and a link of the electronic source. When a user takes a source from library, user will have 30 days to bring it back. If a user doesn’t bring the material back, e-library point of user will decrease (All users will have 100 e-library points when they register the system and one e-library point decreases from e-library points of user for every late day after submission date. When e-point of a user runs out, his/her account will be suspended by the system).

The system will have administrator users. Administrator users will be able to control the system (adding source, deleting source, updating source, controlling the suspended accounts and activating them if necessary etc.).

The system will have different searching types (name of source, type of source). In addition, popular sources will be listed on the website according to popular rented sources. If a user does not enter the system for a year, account of the user will be deleted by the system.

## Development and Execution Environment

Project is developed using Java programming language, HTML programming language and MySQL database. When executing Project, Apache Tomcat 7.0 server is used. Connection between Java and HTML is achieved by using Wicket 1.4.10. In addition MySQL JDBC driver is added to Project for supporting SQL commands.

Project is developed using Eclipse IDE and tested in Windows and Linux operation systems. CSS files added to Project for improving visual beauty of website.

## Tasks of Group Members

### Mustafa Uçar

Number of Student : 040100113

### Tugrul Yatağan

Number of Student : 040100117

### Emre Gökrem

Number of Student : 040100124

### Hüseyin Erdoğan

Number of Student : 040100054

# PART II

# Setup



## Database Creation

## Table Creation

## Connection to MySQL Server

*DatabaseConnection.java* class is used for connecting to database. Method that is used for this purpose is as follows :



*String dbUrl* is used to store surver adress of database.

*String dbUser* is used to store user name.

*String dbPassword* is used to store user password.

All this data is used for connecting to database in this line :

***this****.dbConnect = DriverManager.getConnection(dbUrl,dbUser,dbPassword);*

# PART III

# User Manual



## Guest Interface

### Guest Home Page

Guest will see this page, when he/she enter to website.



### Guest Search Procedure

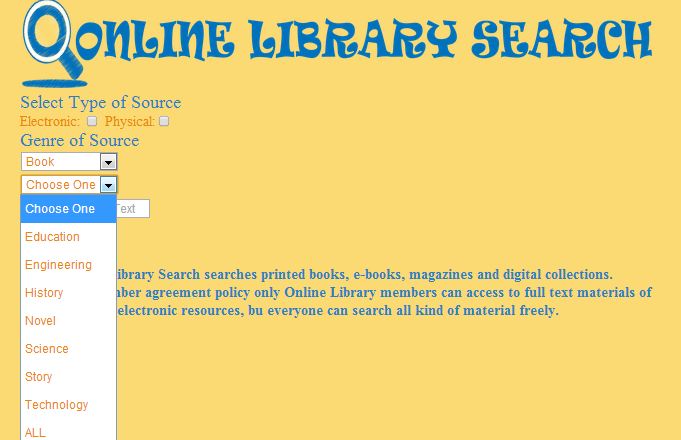
Guest can search using source’s name, type of source(electronic or physical) and genre of source.



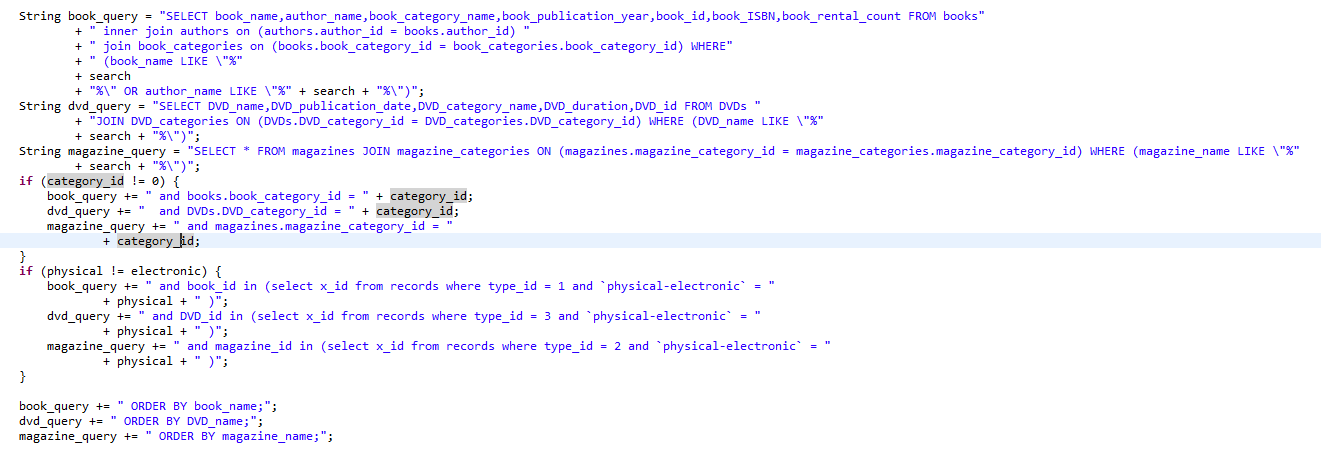
Genre of search part has two dropdown. First drop box have Book, Dvd, Magazine options.



Second drop box have subspecies of selected type of source. For book : education, engineering, history, novel, science, story, technology. For DVD : education, game, movie, song, documantery. For Magazine :comic, computer, engineering, life, science.



Also guest can search book name, author name, dvd name or magazine name. Queries for searching is used in SearchPage.java class. They are like as follows :



After clicking to search button records are showen. Under the Rent row, record have link to record page.



Guest can see information about source but they can not rent sources.



If they want to rent a source, they will see this error message.

Ekran Kırpma

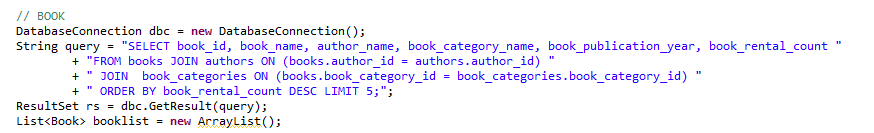
### Most Popular Sources

Guest can see most rented sources in this part.

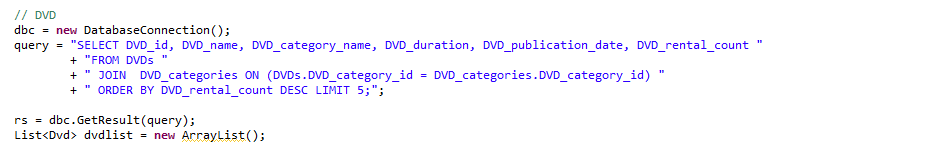


Queries are used for this purpose as follows :

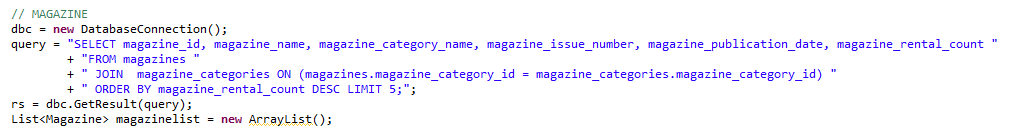
For book :



For dvd :

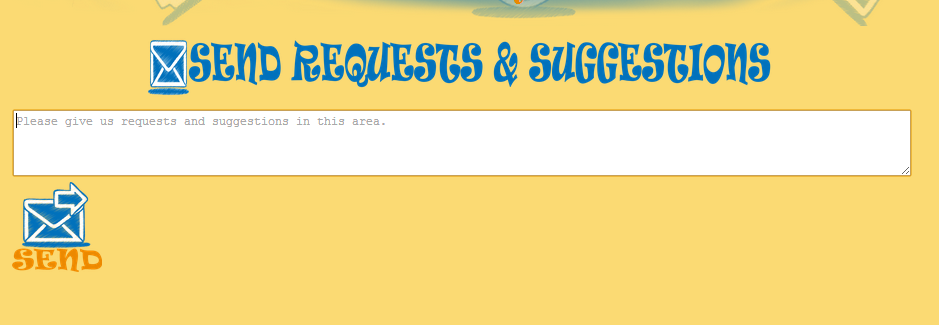


For magazine

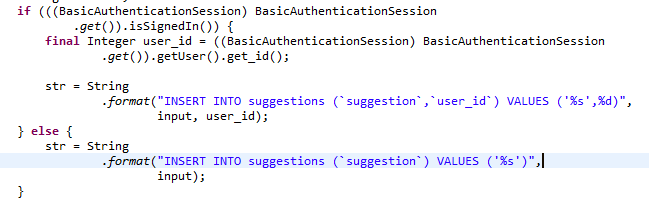


### Request & Suggestions Page

In this part guests can send their requests and suggestions to admin for sake of website. If guests does not register to website, their requests and suggestions will send to admin with “publicuser” nickname.

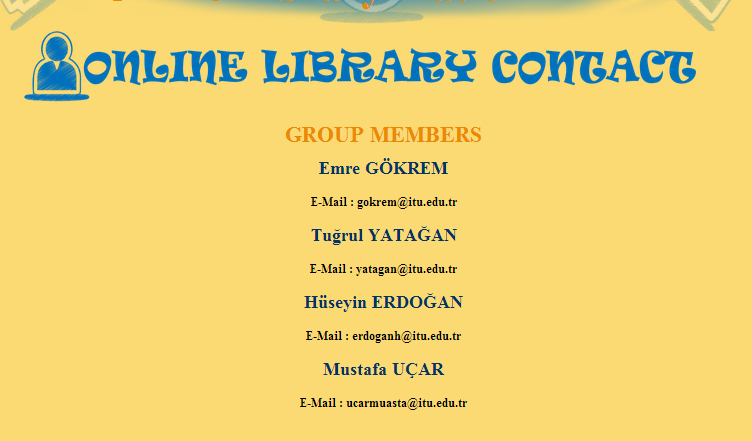


These queries and “suggestions” table are used for requests and suggestions part. Queres are as follows :



### Contact Us Page

In this page group members’s e-mail adresses are showed so guest can interact with us.



### About Us Page

### 

In this page information about project is given. Also informations about libraries that are

used in database and links of them are given.



## User Interface

### Register

Guest can register anytime by clicking register link under login symbol. Guest encounter this

page :



If guest enter another user information, he/she will see this error message.

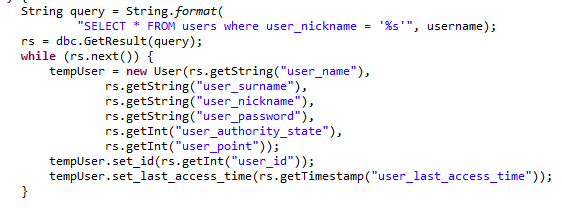
Ekran Kırpma

### Login

After registering, user can login to website using Username and Password.

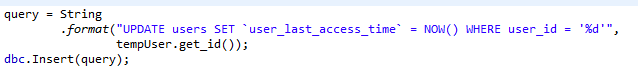


After login we take user informations from database as follows :



According to user\_authority\_state, user is redirected to user or admin homepage. Also user can do everythink that guest can do.

And user\_last\_access\_time is apdated in these lines :



### Logout

User can logout anytime from website by clicking logout icon. Also there is name and

surname under the logout symbol. After Logout user is redirected to guest Home Page .



After Logout user is redirected to guest Home Page .

### User Home Page

User will see this page, when he/she logins to website.



### User Searching Procedure

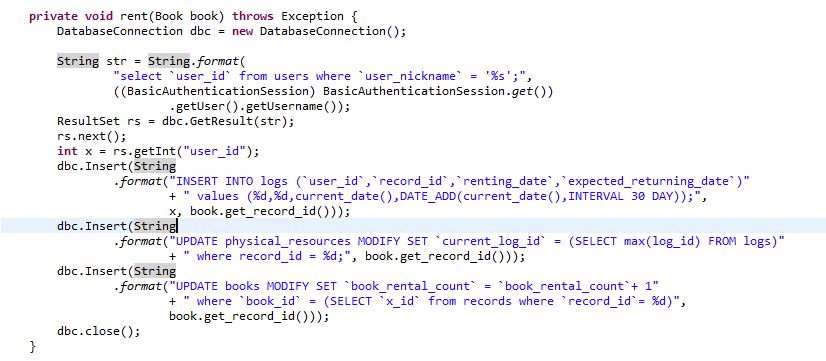
Searching a source is same as guest search procedure. Difference between this two procedure is at source information page. In this page user can rent source which are available. Also user can see location of source and when source is available.



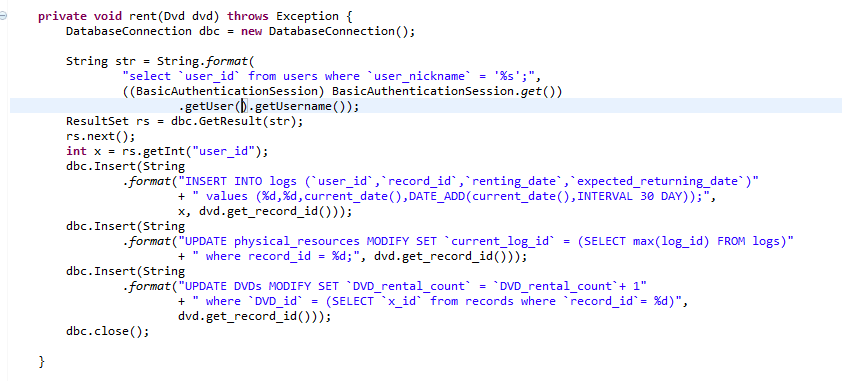
If user press rent link of source that available, information message will be showen. Also user can put sources ,which are not available, to waiting list by clicking Ekran Kırpma link in information page.

For every source type different queries are used for renting part.

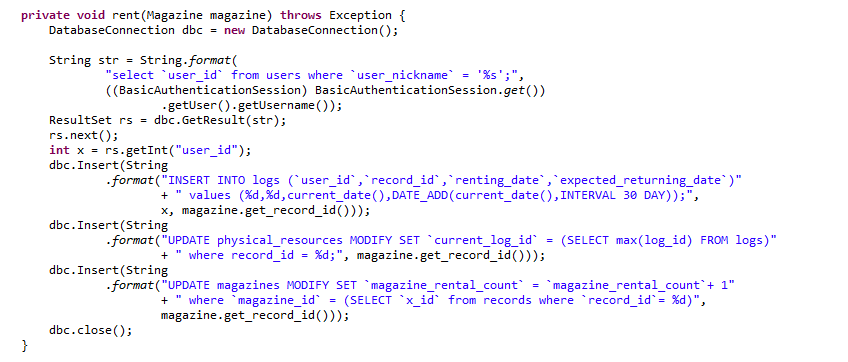
* Queries which are used for renting Book are as follows :



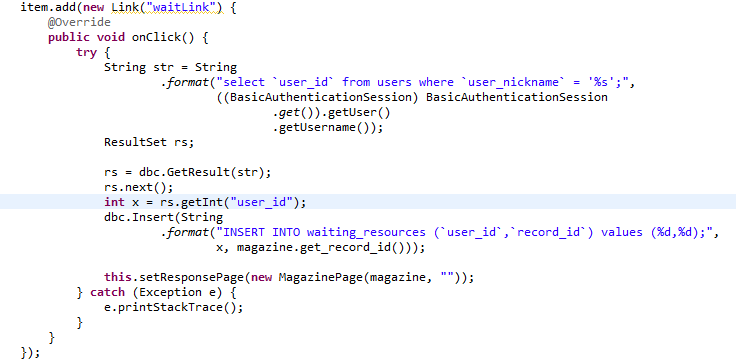
* Queries which are used for renting DVD are as follows :



* Queries which are used for renting Magazine are as follows :



Sources, which are puts to waiting list by user, are stored in “waiting\_resources” table. Queries are as follows.



This part is same for all types.

### Most Popular Page

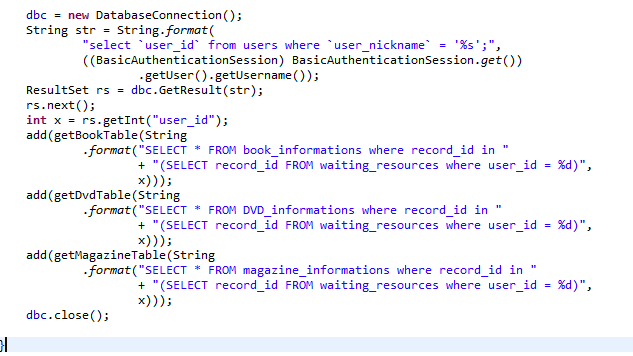
This page is same as guest Most Populars page, except renting procedure. After clicking source’s links user can see location of sources and user can rent them.

### User Profile

User can see firstname, surname, last login time and sources which is user waiting for.



Sources are taken from “waiting\_resources” table in database by using queries as follows :



Also there is a link to suggestion and request page and if user sends request, user name will be seen by admins.

User can delete sources which are in the waiting source list. User\_id and record\_id are used for finding sources.

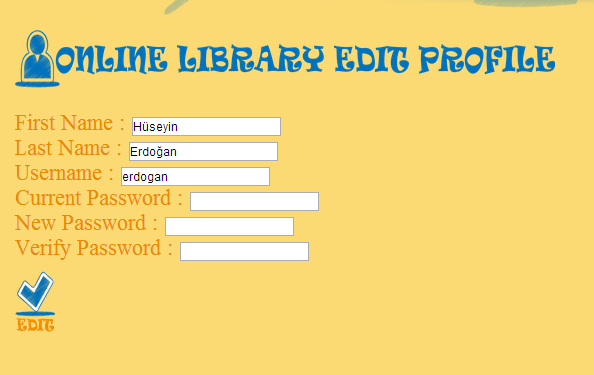


If user clicks Ekran Kırpma button, source which is in waiting list will be deleted. Queries ,which are used for this purpose, are same for all types.

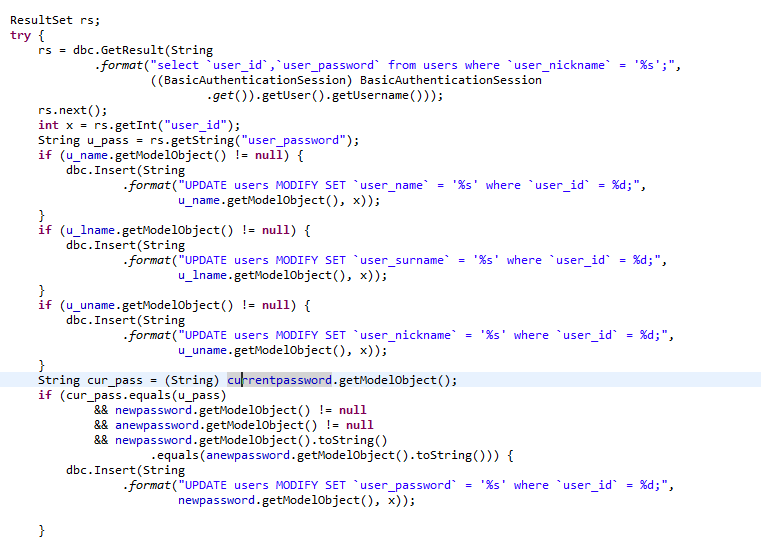


### User Edit Profile

User can change first name, last name, user name and password. Current Password is necessary for changing all of this information.

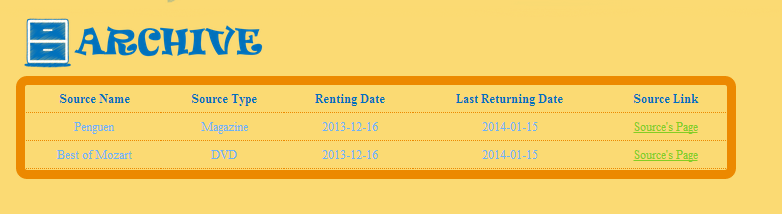


Functions ,that are used in this page, are in UserEditProfileForm.java class. Queries are as follows:

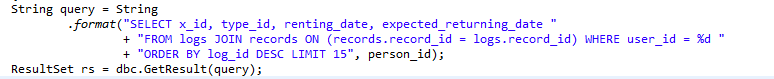


### User Archive

In this page sources are listed which are rented by user.



In this page “logs” table is used for taking sources which are rented by user. Functions ,that are used in this page, are in ArchivePage.java class. Queries are as follows:



## Admin Interface

Admin login to website same as user but different than user admin’s user\_auotority\_state is “1”.

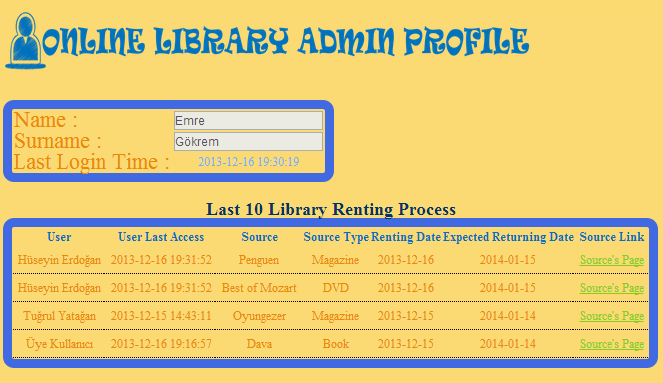
### Admin Home Page

Admin will see this page, when he/she logins to website.

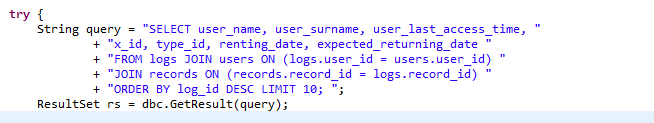


### Admin Profile

Admin will see profil information and last 10 renting process.



Last 10 renting process is used for controlling user activities and functions about this page in AdminProfilePage.java class. Queries, which are used for displaying last 10 renting process, is as follows :



### Admin Edit Profile

This part is same as user edit profile.



### Admin Search Procedure

Search part is same as user search but admins have extra opportunity. Admins can search any item and also they can delete them, update them and put them to libraries. After serching admin will see page like this, which may change according to source type or genre of source.



When admins click to Magazine’s Page, he/she will see this page unlike guest and user.



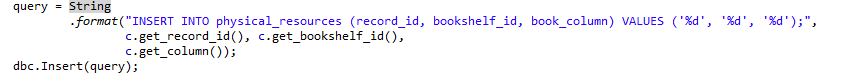
#### Source Placed Page

When admin clicks Ekran Kırpma , he/she will see Online Library Source Placed Page. In this page, admin enters library name and according to library, admin select library floor, bookcase, bookshelf and column number. If source is also electronic, admin can enter source’s URL.

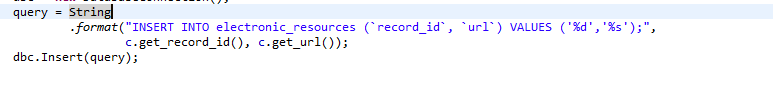


Source Placed Page is common for all tree type.(book, DVD, magazine).

If source is physical, source will be stored in “physical\_resources” table. Queries as follows :



If source is electronic, source will be stored in “electronic \_resources” table. Queries as follows :



#### Source’s Records

In this part user sees location of sources. If user is admin, he/she can edit and delete location of source.



Admin can delete location of source by clicking Ekran Kırpma.

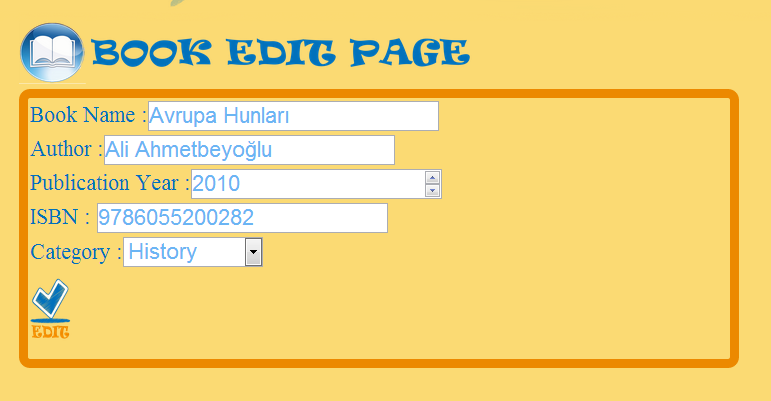
#### Source’s Records Edit

If admin clicks Ekran Kırpma in source’s records table, he/she will see this page. If there is a need update for source’s location, admin can do it in this page.

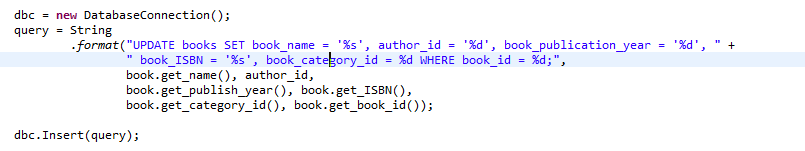


#### Source Edit

When admin clicks Ekran Kırpmain in book information page, he/she will see this page for updating book informations.



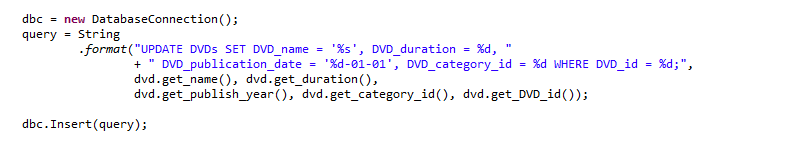
Querie, whic is used in this part as follows :



When admin clicks Ekran Kırpmain in DVD information page, he/she will see this page for updating DVD informations.



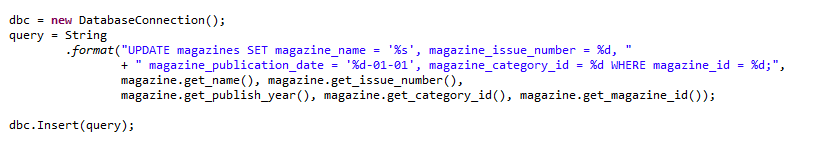
Querie, whic is used in this part as follows :



When admin clicks Ekran Kırpmain in magazine information page, he/she will see this page for updating magazine informations.



Querie, whic is used in this part as follows :

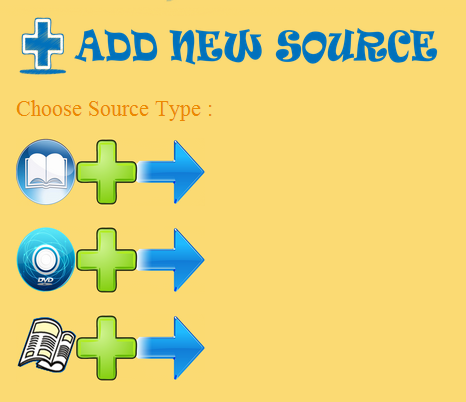


#### Source Delete

When admin clicks Ekran Kırpmafor any type, he/she will be redirected to admin home page and source will be deleted with all of its records.

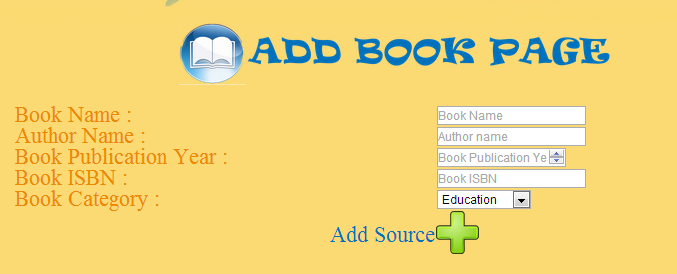
### Adding Source

Admins can add source. First admin selects type of source(book, magazine, dvd).

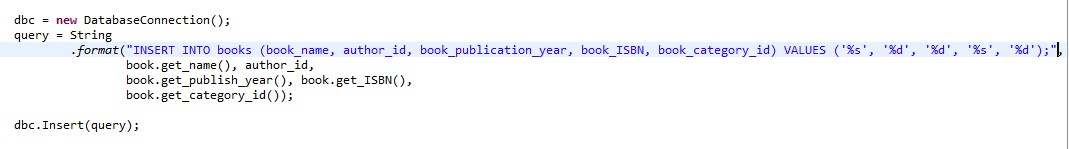


#### Adding Book

After selecting type of source, admin enters information of source. For books : book name, author name, publication year, ISBN and book category.



In this part AddBookForm.java and AddBookPage.java classes are used. Querie is as follows :

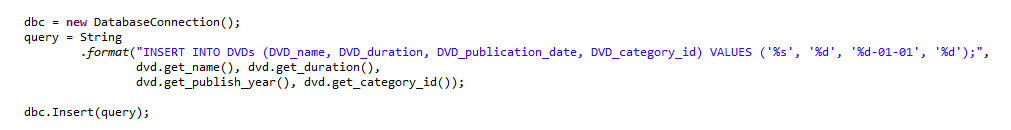


#### Adding DVD

Admin enters information of source. For DVDs : DVD name, publication year, duration and DVD category.



In this part AddDVDForm.java and AddDVDPage.java classes are used. Querie is as follows :

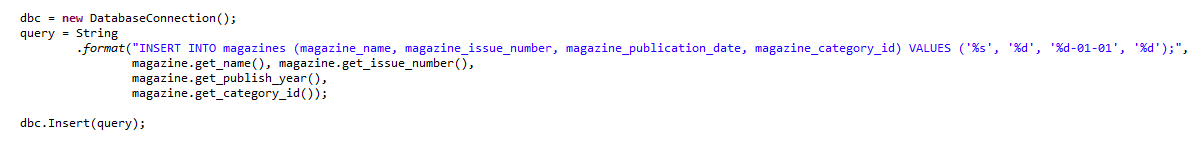


#### Adding Magazine

Admin enters information of source. For magazines : magazine name, publication year, issue number, and magazine category.



In this part AddMagazineForm.java and AddMagazinePage.java classes are used. Querie is as follows :

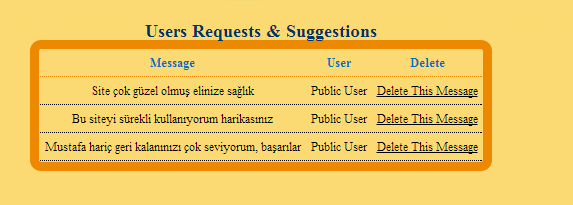


### Admin Archive

Admin can see requests and suggestions in this page.



Also admin can delete this requests and suggestions.



In this part ShowSuggestionsPage.java class is used. Querie is as follows :



If suggestion is send by guest, Public User is written instead of user name.

If suggestion is send by user, user\_name and user\_surname ,which are in “users” table, are used.

# PART IV

# Technical Manual

## Database Design

### Tables

In this chapter tables and some variables are going to be explained.

#### “users” Table

This table stores user data. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| user\_id | İnt(11) | Auto increment | 1 |
| user\_nickname | Varchar(45) | 1 | 0 |
| user\_name | Varchar(45) | 1 | 0 |
| user\_surname | Varchar(45) | 1 | 0 |
| user\_password | Varchar(45) | 1 | 0 |
| user\_authority\_state | bit(1) | 1 | 0 |
| user\_last\_access\_time | Datetime | 1 | 0 |
| user\_point | Varchar(45) | 1 | 0 |

* user\_authority\_state : information about user that is normal user or admin.
* user\_last\_access\_time : last access time of user. This information used for deleting user that does not enter website for over a year.
* user\_point : penalty point. This points are deleted only by admin.

#### “logs” Table

This table stores data about login procedure and sources that are rented. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| log\_id | İnt(11) | 1 | 1 |
| user\_id | İnt(11) | 1 | 0 |
| record\_id | İnt(11) | 1 | 0 |
| renting\_date | date | 1 | 0 |
| expected\_returning\_date | date | 1 | 0 |
| returning\_date | date | 1 | 0 |

* renting\_date : if user rent a source , this shows renting date
* expected\_returning\_date : expected return date of rented source
* user\_authority\_state : return date of rented source

#### “authors” Table

This table stores data about writers of books. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| author\_id | İnt(11) | Auto increment | 1 |
| author\_name | Varchar(45) | 1 | 0 |
| author\_information | mediumtext | 1 | 0 |

* author\_information : Brief introduction about author

#### “books” Table

This table stores book data. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| book\_id | İnt(11) | Auto increment | 1 |
| book\_name | Varchar(45) | 1 | 0 |
| author\_id | İnt(11) | 1 | 0 |
| book\_publication\_year | Year(4) | 1 | 0 |
| book\_ISBN | Varchar(45) | 1 | 0 |
| book\_rental\_count | İnt(11) | 1 | 0 |
| book\_category\_id | İnt(11) | 1 | 0 |

* book\_ISBN : this is for elecronic books
* book\_rental\_count : renting number of books and used for displaying most popular books

#### “magazines” Table

This table magazine data. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| magazine\_id | İnt(11) | Auto increment | 1 |
| magazine\_name | Varchar(45) | 1 | 0 |
| magazine\_issue\_number | İnt(11) | 1 | 0 |
| magazine\_publication\_date | Date | 1 | 0 |
| magazine\_rental\_count | İnt(11) | 1 | 0 |
| magazine\_category\_id | İnt(11) | 1 | 0 |

* magazine\_issue\_number: issue of magazine
* magazine\_rental\_count : renting number of magazines and used for displaying most popular magazines

#### “DVDs” Table

This table stores DVD data. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| DVD\_id | İnt(11) | Auto increment | 1 |
| DVD \_name | Varchar(45) | 1 | 0 |
| DVD \_duration | İnt(11) | 1 | 0 |
| DVD \_publication\_date | Date | 1 | 0 |
| DVD \_rental\_count | İnt(11) | Default “0” | 0 |
| DVD \_category\_id | İnt(11) | 1 | 0 |

* DVD\_rental\_count : renting number of DVDs and used for displaying most popular DVDs

#### “book\_categories” Table

This table stores data about book categories. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| book\_categorie\_id | İnt(11) | Auto increment | 1 |
| book\_categorie\_name | Varchar(45) | 1 | 0 |

* book\_categorie\_name : used for categorizing books. Ex : bilim, eğitim, roman.

#### “bookcases” Table

This table stores data about bookcases and this informations are used for locating books. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| bookcase\_id | İnt(11) | Auto increment | 1 |
| floor \_id | İnt(11) | 1 | 0 |
| bookcase\_name | Varchar(10) | 1 | 0 |

* floor \_name : floor name that is bookcase in there

#### “bookshelves” Table

This table stores data about bookshelves and this informations are used for locating bookcase. Also bookshelves have limit. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| ookshelf\_id | İnt(11) | Auto increment | 1 |
| bookcase\_id | İnt(11) | 1 | 0 |
| ookshelf \_name | Varchar(10) | 1 | 0 |
| book\_limit | İnt(11) | 1 | 0 |

* bookcase\_id : bookcase id that is ookshelf in there
* book\_limit : maximum number of book that ‘s Bookshelf contains

#### “DVD\_categories” Table

This table stores data about categories of DVDs. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| DVD\_category\_id | İnt(11) | Auto increment | 1 |
| DVD\_category\_name | Varchar(45) | 1 | 0 |

* DVD\_category\_name : used for categorizing DVDs. Ex : dizi, film, eğitim.

#### “electronic\_resources” Table

This table stores data about sources that are electronic. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| record\_id | İnt(11) | Auto increment | 1 |
| url | Varchar(45) | 1 | 0 |

* url : users can reach sources on internet using url

#### “physical\_resources” Table

This table stores data about sources that are physical. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| record\_id | İnt(11) | Auto increment | 1 |
| bookshelf \_id | İnt(11) | 1 | 0 |
| book \_column | İnt(11) | 1 | 0 |
| current\_log\_id | İnt(11) | 1 | 0 |

#### “waiting\_resources” Table

This table stores data about users that are waiting for sources. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| user\_id | İnt(11) | 1 | 1 |
| record\_id | İnt(11) | 1 | 1 |

* record\_id : source that is users waiting for

#### “floors” Table

This table stores data about floors that libraries have. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| floor\_id | İnt(11) | Auto increment | 1 |
| library\_id | İnt(11) | 1 | 1 |
| floor\_name | Varchar(45) | 1 | 1 |

* library\_id : library that’s contains that floor

#### “libraries” Table

This table stores library data. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| library\_id | İnt(11) | Auto increment | 1 |
| library\_name | Varchar(45) | 1 | 0 |
| library\_address | Varchar(45) | 1 | 0 |

#### “magazine\_categories” Table

This table stores data about magazine categories. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| magazine\_categorie\_id | İnt(11) | Auto increment | 1 |
| magazine\_categorie\_name | Varchar(45) | 1 | 0 |

* magazine\_categorie\_name : used for categorizing magazines. Ex : bilim, eğitim ,mühendislik

#### “records” Table

This table stores data about sources and type of this sources. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| record\_id | İnt(11) | Auto increment | 1 |
| type\_id | İnt(11) | 1 | 0 |
| x\_id | İnt(11) | 1 | 0 |
| Physical\_electronic | bit(1) | 1 | 0 |

* type\_id : type of record. Ex : book,DVD, magazine
* physical-electronic : record is physical or electronic

#### “suggestions” Table

This table stores suggestions that are going to send to admin. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| suggestion\_id | İnt(11) | 1 | 1 |
| suggestion | text | 1 | 0 |
| User\_id | İnt(11) | 1 | 0 |

#### “types” Table

This table stores data about source type. Informations in this table as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Not Null** | **Primary Key** |
| type\_id | İnt(11) | 1 | 1 |
| type\_name | Varchar(45) | 1 | 0 |

* type\_name : contains types. Ex: book, DVD, magazine

### Views

In this chapter views are going to be displayed. Also usage way of views are going to be explained.

In source page this views are used.

#### “book\_information” View

This view is used when just book is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| book\_name | Varchar(45) |
| author\_name | Varchar(45) |
| book\_publication\_year | Year(4) |
| book\_ISBN | Varchar(45) |
| book\_rental\_count | İnt(11) |
| library\_name | Varchar(45) |
| floor\_name | Varchar(10) |
| bookcase\_id | İnt(11) |
| bookshelf\_id | İnt(11) |
| book\_column | İnt(11) |
| physical\_electronic | bit(1) |
| available | İnt(11) |
| url | Varchar(45) |

#### “DVD\_information” View

This view is used when just DVD is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| DVD\_id | İnt(11) |
| DVD\_name | Varchar(45) |
| physical\_electronic | bit(1) |
| DVD\_duration | İnt(11) |
| DVD\_publication\_date | date |
| DVD\_rental\_count | İnt(11) |
| available | İnt(7) |

#### “electronic\_books” View

This view is used when book and electronic is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| book\_name | Varchar(45) |
| author\_name | Varchar(45) |
| book\_publication\_year | Year(4) |
| book\_ISBN | Varchar(45) |
| book\_rental\_count | İnt(11) |
| physical\_electronic | bit(1) |
| url | Varchar(45) |
| available | İnt(7) |

#### “electronic\_magazines” View

This view is used when magazine and electronic is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| magazine\_name | Varchar(45) |
| magazine\_publication\_date | date |
| magazine\_issue\_number | İnt(11) |
| magazine\_rental\_count | İnt(11) |
| physical\_electronic | bit(1) |
| url | Varchar(45) |
| available | İnt(7) |

#### “magazine\_informations” View

This view is used when just magazine is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| magazine\_name | Varchar(45) |
| magazine\_publication\_date | date |
| magazine\_issue\_number | İnt(11) |
| magazine\_rental\_count | İnt(11) |
| library\_name | Varchar(45) |
| floor\_name | Varchar(10) |
| bookcase\_id | İnt(11) |
| bookshelf\_id | İnt(11) |
| book\_column | İnt(11) |
| physical\_electronic | bit(1) |
| Available | İnt(11) |
| url | Varchar(45) |

#### “physical\_books” View

This view is used when book and physical is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| book\_name | Varchar(45) |
| author\_name | Varchar(45) |
| book\_publication\_year | Year(4) |
| book\_ISBN | Varchar(45) |
| book\_rental\_count | İnt(11) |
| library\_name | Varchar(45) |
| floor\_name | Varchar(10) |
| bookcase\_id | İnt(11) |
| bookshelf\_id | İnt(11) |
| book\_column | İnt(11) |
| physical\_electronic | bit(1) |
| Available | İnt(11) |

#### “physical\_magazines” View

This view is used when magazine and physical is selected in searce page.

|  |  |
| --- | --- |
| **Name** | **Type** |
| record\_id | İnt(11) |
| magazine\_name | Varchar(45) |
| magazine\_publication\_date | date |
| magazine\_issue\_number | İnt(11) |
| magazine\_rental\_count | İnt(11) |
| library\_name | Varchar(45) |
| floor\_name | Varchar(10) |
| bookcase\_id | İnt(11) |
| bookshelf\_id | İnt(11) |
| book\_column | İnt(11) |
| physical\_electronic | bit(1) |
| Available | İnt(11) |

### Foreign Keys

#### In “bookscases” table

floor\_id references floor\_id in “floors” table

* On delete cascade
* On update cascade

#### In “bookshelves” table

bookcase\_id references bookcase\_id in “bookcase” table.

* On delete cascade
* On update cascade

#### In “electronic\_resources” table

record\_id references record\_id in “records” table.

* On delete cascade
* On update cascade

#### In “floors” table

library\_id references library\_id in “libraries” table.

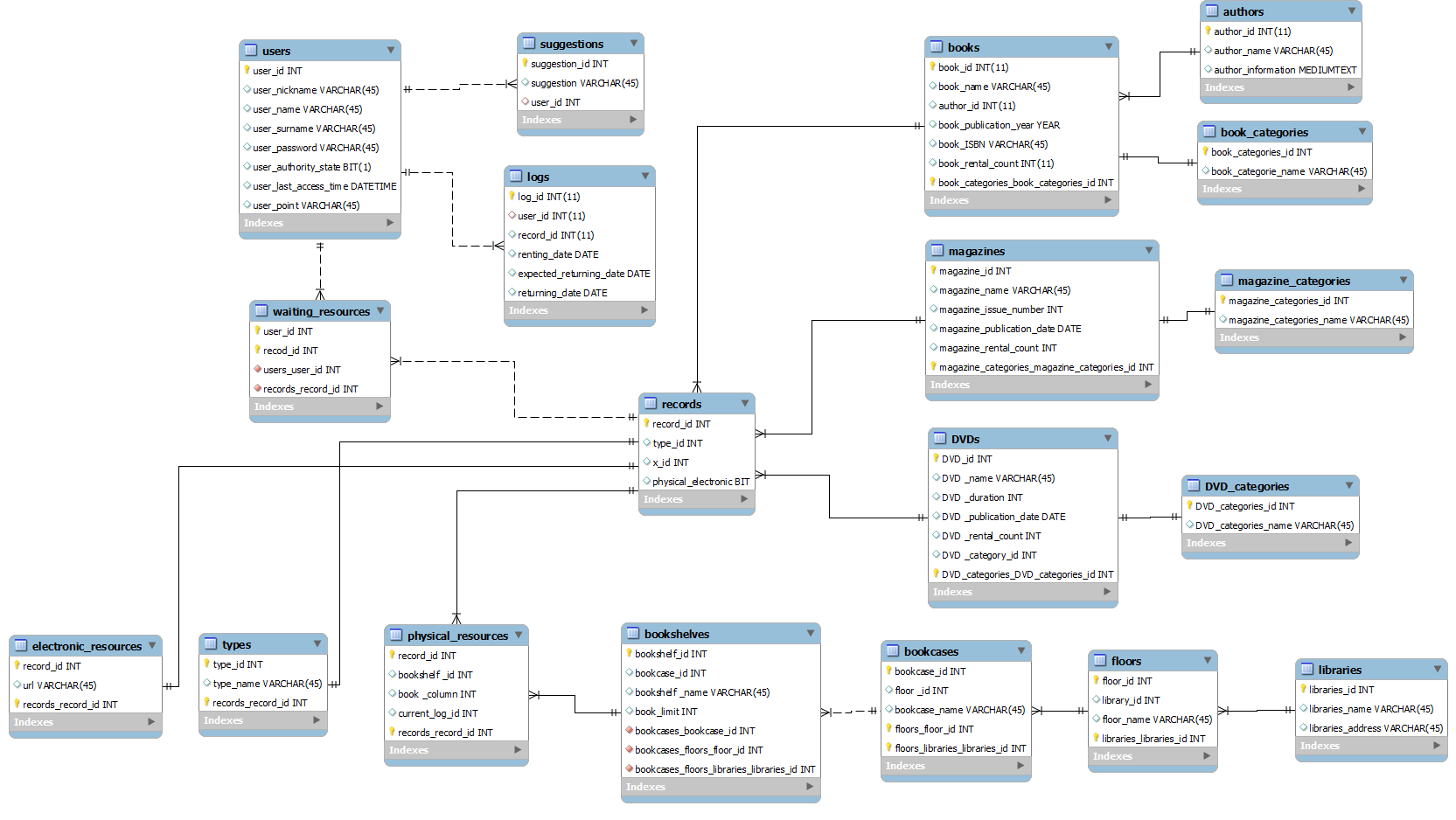
* On delete cascade
* On update cascade

#### In “physical\_resources” table

record\_id references record\_id in “records” table.

* On delete cascade
* On update cascade

### Entity Relationship Diagram



## Software Design

### Admin Classes

### User Classes

### Book Classes

### Magazine Classes

### DVD Classes

### Additional Classes