



**Bilkent University**  
**CS 353**  
**Final Report**

**Group 26**  
**School Library Database**

Cemhan Kaan Özaltan - 21902695 - Section 1  
Hissam Mahmoud Elsayed Faramawy - 21901253 - Section 3  
Servet Gülnarođlu - 21902474 - Section 2  
Taha Batur Őenli - 21901857 - Section 3

## **Table of Contents**

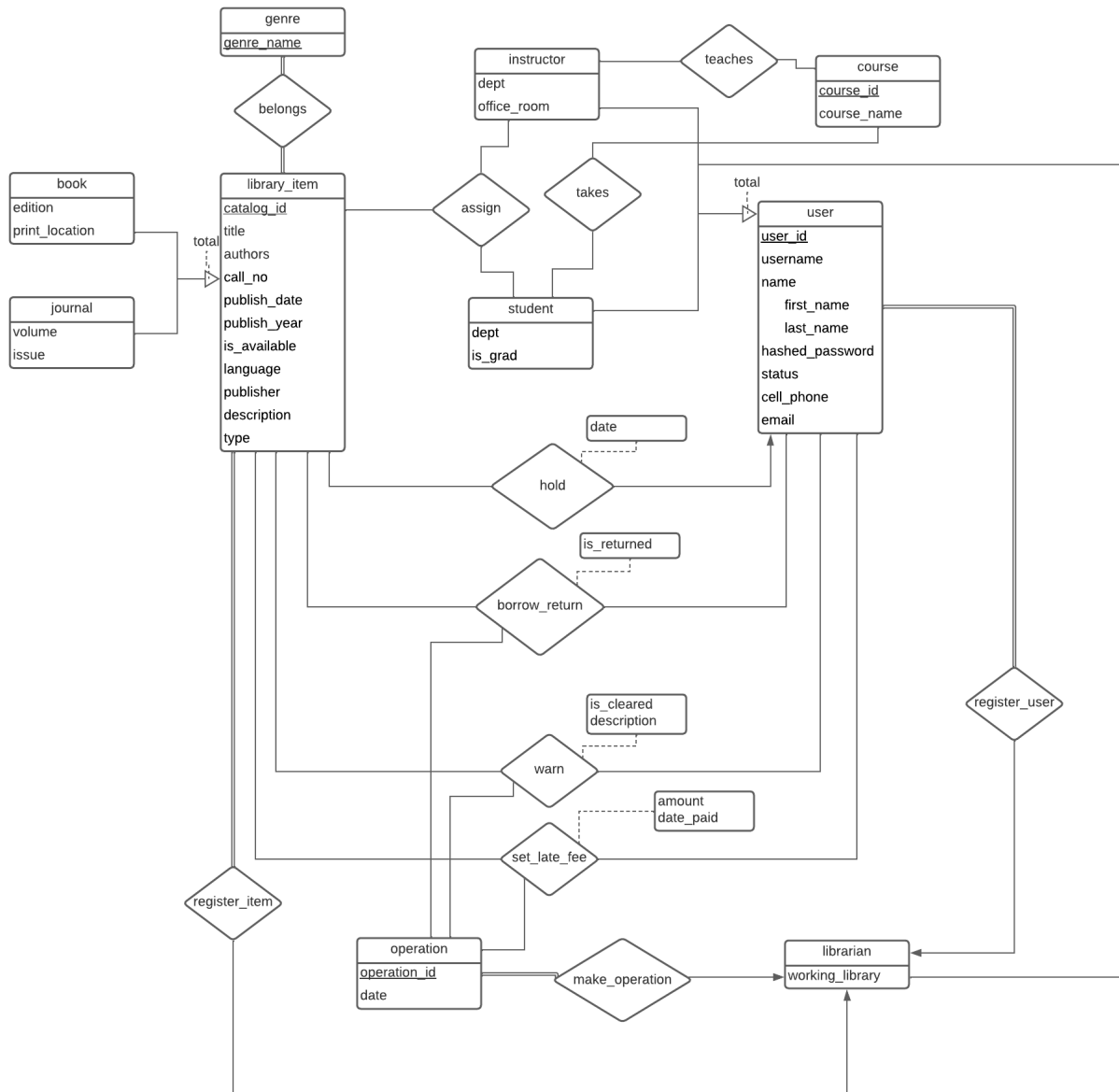
<b>1. Application System Description</b>	<b>3</b>
<b>3. Final ER Diagram</b>	<b>4</b>
<b>4. Final Tables</b>	<b>4</b>
<b>5. Implementation Details</b>	<b>6</b>
<b>6. Advanced Database Features</b>	<b>7</b>
<b>6.1. Reports</b>	<b>7</b>
<b>6.2. Views</b>	<b>8</b>
<b>6.3. Triggers</b>	<b>9</b>
<b>6.4. Constraints</b>	<b>9</b>
<b>6.5. Stored Procedures</b>	<b>10</b>
<b>7. User's Manual</b>	<b>10</b>
<b>7.1. Login Page</b>	<b>12</b>
<b>7.2. Register Student</b>	<b>13</b>
<b>7.3. Register Instructor</b>	<b>14</b>
<b>7.4. Student Main Page</b>	<b>15</b>
<b>7.5. Instructor Main Page</b>	<b>15</b>
<b>7.6. Librarian Main Page</b>	<b>15</b>
<b>7.7. Borrow/Return Item Page</b>	<b>16</b>
<b>7.8. Hold Item Page</b>	<b>16</b>
<b>7.9. Assigned Items Page and Assign Items Page</b>	<b>17</b>
<b>7.10. Register Library Item</b>	<b>18</b>
<b>7.11. Warn User Page</b>	<b>18</b>
<b>7.12. View My Items Page</b>	<b>19</b>
<b>7.13. View Warnings</b>	<b>20</b>
<b>7.14. Add Course Page</b>	<b>20</b>
<b>8. Website</b>	<b>21</b>
<b>9. References</b>	<b>21</b>

## 1. Application System Description

This project is an online database to satisfy different needs of a university library such as borrowing library items, adding new library items and removing existing ones. The system has three user types: students, instructors and librarians. All users have a unique id, name, password, status, cell phone number and email saved in the system. All user types have a few additional attributes of their own. Users will use their ID and password to login to the system. Students, through librarians, can borrow or return library items and also pay their late return fees for items that they didn't return in time. Meanwhile instructors, can assign library items to students, borrow and return library items and also pay their late return fees. Finally, the librarians will act as administrators and handle tasks such as registering and removing library items, warning users and charging late return fees. All of these actions are considered as operations with unique IDs and different types for categorization. Librarians are also able to list all existing users except librarians, along with applying several filters to this search.

Each library item in the system will have a unique ID, title, author or authors, call number, date of publication, year of publication, availability status, language, publisher and description. There are two types of library items, books and journals. Books have edition and print location as extra values while journals have volume and issue. The library items can be listed using combinations of several filters on different attributes by all user types.

### 3. Final ER Diagram



Note: Here, the authors attribute of library\_item is changed from multivalued to single-valued. Also, the name of the relationship set between operation and librarian is changed from create to make\_operation. These changes along with a few varchar size changes are included in the final tables.

### 4. Final Tables

library\_item(catalog\_id: int, title: varchar(128), call\_no: varchar(128), publish\_date: date, publish\_year: int, is\_available: boolean, language: varchar(64), type: varchar(64), publisher: varchar(64), description: varchar(256), authors: varchar(128))

genre(genre\_name: varchar(64))

belongs(catalog\_id: int, genre\_name: varchar(64))

catalog\_id: Foreign key to library\_item

genre\_name: Foreign key to genre

book(catalog\_id: int, edition: int, print\_location: varchar(64))

catalog\_id: Foreign key to library\_item

journal(catalog\_id: int, volume: int, issue: int)

catalog\_id: Foreign key to library\_item

user(user\_id: int, username: varchar(64), first\_name: varchar(64), last\_name: varchar(64), hashed\_password: varchar(64), status: boolean, cell\_phone: varchar(12), email: varchar(64))

instructor(user\_id: int, dept: varchar(64), office\_room: varchar(64))

user\_id: Foreign key to user

student(user\_id: int, dept: varchar(32), is\_grad: boolean)

user\_id: Foreign key to user

librarian(user\_id: int, working\_library: varchar(64))

user\_id: Foreign key to user

operation(operation\_id: int, date: date)

hold(catalog\_id: int, user\_id: int, date: date)

catalog\_id: Foreign key to library\_item

user\_id: Foreign key to user

borrow\_return(catalog\_id: int, operation\_id: int, user\_id: int, is\_returned: boolean)

catalog\_id: Foreign key to library\_item

operation\_id: Foreign key to operation

user\_id: Foreign key to user

warn(catalog\_id: int, operation\_id: int, user\_id: int, description: varchar(256), is\_cleared: boolean)

user\_id: Foreign key to user

catalog\_id: Foreign key to library\_item

operation\_id: Foreign key to operation

set\_late\_fee(catalog\_id: int, operation\_id: int, user\_id: int, amount: int, date\_paid: date)

user\_id: Foreign key to user

catalog\_id: Foreign key to library\_item

operation\_id: Foreign key to operation

course(course\_id: int, course\_name: varchar(20))

teaches(course\_id: int, user\_id: int)  
 course\_id: Foreign key to course  
 user\_id: Foreign key to user

takes(course\_id: int, user\_id: int)  
 course\_id: Foreign key to course  
 user\_id: Foreign key to user

assign(catalog\_id: int, student\_user\_id: int, instructor\_user\_id: int)  
 catalog\_id: Foreign key to library\_item  
 student\_user\_id: Foreign key to student(user\_id)  
 instructor\_user\_id: Foreign key to instructor(user\_id)

make\_operation(operation\_id: int, user\_id: int)  
 operation\_id: Foreign key to operation  
 user\_id: Foreign key to user

## 5. Implementation Details

In our implementation, we used MySQL as our database management system, Node.js for the backend and React.js for our frontend. We used MySQL Workbench to set up a local server, in which we created a database for our website. We implemented Javascript scripts for creating all tables according to our design report, with a few changes. We also added scripts for purposes like creating the admin users (Librarians) and dropping certain tables for testing. For our backend, we used the mysql package of Node.js to first connect to our database using its credentials and a connection object. We then used this same package to process our SQL queries and statements using several backend functions for purposes like registering users and library items and searching them. We used a singleton object which fulfilled a similar role as a token in order to store information gathered from the database about a user and use it throughout a single session. We implemented our frontend according to the mockup diagrams in our design reports with a few changes, mainly using the Material UI package of React.js along with a few others installed through npm. Inputs and displays are done through these components.

Throughout our implementation, we encountered several difficulties at different stages. The implementation and usage of the database in MySQL, along with connecting to the database to our backend was fairly easy since all members had prior knowledge on SQL. We only had difficulties when searching users and items using several criteria, which we solved using regular expressions. Backend and frontend implementation was slightly

more difficult since not all members had experience with the used frameworks. For our backend, processing queries through code was a slight problem, which we solved by using the mysql package as mentioned. We encountered problems like out of margin text fields and incompatible components in the frontend, which we solved by searching for and using more appropriate components.

### Group Member Contributions

Cemhan Kaan Özaltan: Setting up the database, backend functionalities (inserting library items and users, search features and queries, advanced database features).

Servet Gülnaroğlu: All frontend functionality, item operations, hold, borrow and return functionalities, librarian features, librarian warnings to users, overall system integration, front-backend connection.

Taha Batur Şenli: Worked on backend functionalities involving user types like instructor and their specific features (display assigned library items, assign library items etc.)

Hissam Mahmoud Elsayed Faramawy: Worked on some extra functionalities to backend and corresponding frontend implementations.

## 6. Advanced Database Features

### 6.1. Reports

#### Total Number of Library Items for Each Genre

```
SELECT genre_name, count(*) AS cnt
FROM library_item NATURAL JOIN belongs NATURAL JOIN genre
GROUP BY genre_name;
```

#### Borrow Counts of Each Library Item Which Has Been Borrowed At Least Once in Alphabetical Title Order

```
WITH total_borrows AS (
    SELECT catalog_id, count(*) as borrow_cnt
    FROM borrow_return
    GROUP BY catalog_id)
SELECT title, catalog_id
FROM total_borrows NATURAL JOIN library_item
ORDER BY title ASC;
```

UI view of these reports is as follows:

## Reports

Total Number of Library Items For Each Genre	
Genre Name	Item count
Horror	10
Fiction	8
Dram	3

Borrow Counts of Each Library Item		
Title	Catalog ID	Total Borrow Count
Lord of the Rings	31	2
AI	1003	8
Martin Eden	32	2
Chess	33	100

## 6.2. Views

### Users View for Librarians (User Search Page)

Librarians will use the following view to see all users since they should not see login credentials.

```
CREATE VIEW users_for_librarian AS
SELECT user_id, username, first_name, last_name, status, cell_phone, email
FROM user;
```

### Users With Restricted Status for Librarians (User Search Page)

Librarians will use this view to access students and instructors who are currently not able to borrow books until they pay their fees.

```
CREATE VIEW restricted_users AS
SELECT *
FROM user
WHERE status = false;
```

### List of Currently Borrowed Items for Librarians (Item Search Page for Librarian)

Currently borrowed items will be listed with this view in order to make the librarian's task of detecting and setting late fees quicker.

```
CREATE VIEW currently_borrowed AS
```



```
SELECT catalog_id
FROM borrow_return
WHERE is_returned = false;
```

### 6.3. Triggers

The following triggers perform automatic operations that preserve the consistency of our system

#### Set Users With Late Fees as Restricted Users

```
DELIMITER //
CREATE TRIGGER set_restricted AFTER INSERT ON set_late_fee
FOR EACH ROW
BEGIN
    UPDATE user
    SET status = false
    WHERE NEW.user_id = user.user_id;
END //
DELIMITER ;
```

#### Set Users Who Paid Their Fees to Clear Status

```
DELIMITER //
CREATE TRIGGER remove_restricted AFTER UPDATE ON set_late_fee
FOR EACH ROW
BEGIN
    UPDATE user
    SET status = true
    WHERE NEW.user_id = user.user_id;
END //
DELIMITER ;
```

### 6.4. Constraints

1. Users must register to the system through a librarian in order to use it.
2. Passwords are at most 20 and at least 6 characters long.
3. A user can put 5 items on hold at a time.
4. A user can borrow 5 items at a time.
5. Users must pay their pending late fees if they are in the restricted status in order to make further holds or borrows. Otherwise, they will be unable to use the system.

6. An item cannot be held, borrowed or returned at the same time more than once (logical error otherwise, must be returned to be borrowed again and vice versa).
7. All fields must be filled when registering a user by a librarian (not the case for items as some details may be unknown or non-existent).

Other constraints are specified in the table creation statements, such as primary keys, foreign keys and conditions on what to do when a foreign key is updated or deleted, along with not null values.

## 6.5. Stored Procedures

The following stored procedures will be used due to the frequent need for them.

### Login Check

Returns a table with 1 row if login successful, no rows if else.

```
DELIMITER //
CREATE PROCEDURE login(IN uid int, IN pass varchar(20)) BEGIN
    SELECT *
    FROM user
    WHERE user_id = uid AND hashed_password = pass;
END //
DELIMITER ;
```

### Browse Library Items by Title

Most common search method, with the only filter being the title of the item.

```
DELIMITER //
CREATE PROCEDURE search_by_title(IN search_title varchar(20)) BEGIN
    SELECT *
    FROM library_item
    WHERE title = search_title;
END //
DELIMITER ;
```

## 7. User's Manual

The Graphical User Interface (GUI) of our application displays certain actions undertaken by a multitude of users. We have a universal login page for all types of users (see Subheading 7.1) which are student, librarian and instructor. Using this universal login page all users can enter the system with the provided user ID and password. Librarian user type is created by the developers and their password is

provided beforehand. Furthermore, other types of users can't sign up to the system and instead a librarian registers accounts for both students and instructors (see Subheadings 7.2 and 7.3).

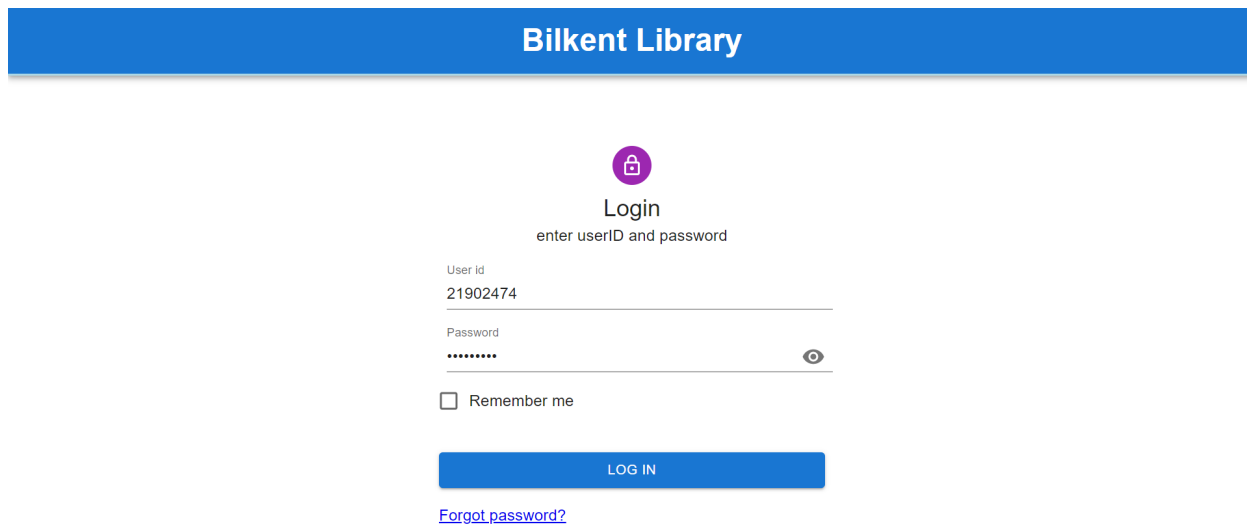
Once the user successfully logs into the system, they are directed to a main page which contains detailed information about that user, called the main page (see Subheadings 7.4, 7.5 and 7.6). Please note that the contents of this main page varies from one user type to the other, because all user types have certain unique attributes. The login procedure is checked by using an SQL query that looks at the entered user ID and password to see if there is a match. Similarly, an SQL statement is executed to complete the registry process, in which the information provided by the librarian is inserted into the database. This insert operation is different for a student and instructor.

Users can borrow available library items or return the ones they have already borrowed. When the user wants to return or borrow an item, they are directed to a page through which they see all available library items and their status. Items are available, borrowed by another user or borrowed by the logged in user (see Subheading 7.7). From this page users can also hold free or borrowed books so that they can pick it up at the first opportunity or when they are returned to the library (see Subheading 7.8). Users can also search for a specific book through this page using the search bar. Additionally, instructors can assign certain books to certain students via this page. To do so, they first select the "assign to students" option on a certain book (see Subheading 7.9).

Alongside registering instructors and students, librarians can also register new library items to the library database or delete existing items (see Subheading 7.10). Librarians can also warn users that their book is soon due to be returned to the library or, if the book hasn't been returned past its deadline, warn the users that they have passed the deadline and incurred a certain late fee. Either way, the users receive a notification from the library system that they have incurred a warning and, if the item is overdue, a fee (see Subheading 7.11). Finally, librarians can set the mentioned late fee's amount for users who haven't returned their books.

Additionally, users can also see the items they have borrowed from the "My Items" page (see Subheading 7.12). Finally, they can also see their warnings from the "Warnings" page (see Subheading 7.13). Also, students can access their enrolled courses and enroll in courses using our program (see Subheading 7.14).

## 7.1. Login Page




The screenshot shows the login page for Bilkent Library. At the top, there is a blue header with the text "Bilkent Library". Below the header, there is a purple lock icon, followed by the word "Login" and the instruction "enter userID and password". There are two input fields: "User id" with the value "21902474" and "Password" with a masked password ".....". A "Remember me" checkbox is present below the password field. A blue "LOG IN" button is located below the "Remember me" checkbox. At the bottom, there is a blue link labeled "Forgot password?".

From the page in the above figure, users can log in by typing their names and passwords in the specified sections then click Log In button. They can also click “Remember me” beforehand to remember them next time they use the website. Moreover, they can select “Forgot password” to to retrieve their passwords.

## 7.2. Register Student

### Bilkent Library

student LIBRARY ITEMS USERS REGISTER A NEW ITEM REGISTER A NEW USER



#### Register A User

Select user type

**STUDENT** INSTRUCTOR

Email  
servetggg@gmail.com

Bilkent ID  
21902473

First name  
Servet

Last name  
Gulnaroglu

User name  
servetg

Phone Number  
05051461111

Department

Is Graduate?


REGISTER

From this page, Librarian user can register a student by first selecting “STUDENT” from the top of the form and then entering the student’s “email”, “Bilkent ID”, “First Name”, “Last Name”, “User name”, “Phone Number”, and “Department”. They also can select “is Graduate?” to specify that they are a graduate student. After entering the necessary information, they click “REGISTER” to finalize the student registration process.

### 7.3. Register Instructor

**Bilkent Library**

student LIBRARY ITEMS USERS REGISTER A NEW ITEM REGISTER A NEW USER

 Register A User

Select user type

STUDENT INSTRUCTOR

Email  
servetggg@gmail.com

Bilkent ID  
21902473

First name  
Servet

Last name  
Gulnaroglu

User name  
servetg

Phone Number  
05051461111

Office Number  
EA-506

Department

REGISTER

From this page, Librarian user can register a instructor by first select “INSTRUCTOR” from the top of the form and then entering the instructor’s “email”, “Bilkent ID”, “First Name”, “Last Name”, “User name”, “Phone Number”, “Office Number”, and “Department”. After entering the necessary information, they click “REGISTER” to finalize the instructor registration process.

## 7.4. Student Main Page



## 7.5. Instructor Main Page



## 7.6. Librarian Main Page



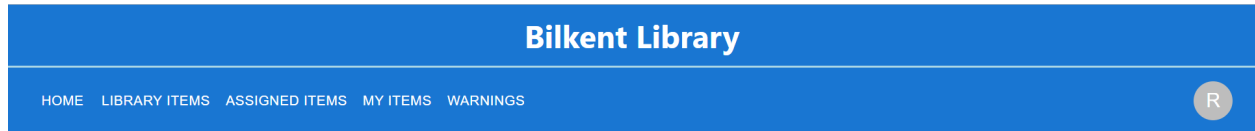
ALL STUDENT INSTRUCTOR

User ID	Last Name	
1	Cinel	SELECT
434	Guinaroglu	SELECT
3131	Guinaroglu	SELECT
21902474	Guinaroglu	SELECT

This page is the main page for Librarian user. Librarian user can see all borrowed items from this page. The can also search a user by their id or last name. Moreover, instead of viewing “ALL” users, they can filter them to “STUDENT” or “INSTRUCTOR” users only.

They can also select other pages from the top section: “LIBRARY ITEMS”, “USERS”, “REGISTER A NEW ITEM”, “REGISTER A NEW USER”.

## 7.7. Borrow/Return Item Page



Title	Author	Year	Type	Status
Advances in artificial intelligence: applications and theory	Bezdek, James C.	2019	BOOK	Borrowed
Advances in artificial intelligence: applications and theory	Bezdek, James C.	2019	BOOK	Borrowed

Using this panel the user can access the items and borrow or return them.

## 7.8. Hold Item Page

The screenshot shows the "Hold Item Page" of the Bilkent Library system. At the top, there is a blue navigation bar with the text "Bilkent Library" and a navigation menu with links: "HOME", "LIBRARY ITEMS", "ASSIGNED ITEMS", "MY ITEMS", and "WARNINGS". On the right side of the navigation bar, there is a circular icon with the letter "R".

A success message is displayed in the center: "localhost:3000 şunu diyor: Succesfully holded!". Below the message is a blue button labeled "Tamam".

Below the message, there is a table with the following data:

Title	Author	Year	Type	Status
Advances in artificial intelligence: applications and theory	Bezdek, James C.	2019	BOOK	Borrowed
Advances in artificial intelligence: applications and theory	Bezdek, James C.	2019	BOOK	Borrowed
sda	Bezdek, James C.	2019	BOOK	HOLD
fgs	Bezdek, James C.	2019	BOOK	HOLD



From this page, Student or Instructor user can hold a book by selecting “HOLD” button. Please notice that they can only hold the book if it is available. If the book is held successfully, a message will pop-up showing a success message.

## 7.9. Assigned Items Page and Assign Items Page



### Items assigned by your instructors

Name	Type	Instructor
Beyaz Bruhlar Ulkesinde Kaan Ozaltan published at 2019	Book	Coker A.
Selamun aleykum Hiassam published at 2019	Book	Cancak Russo

From this page, students can view items assigned by their instructor. They can also see an item's type and the instructor who assigned it.



Bilkent Library				
HOME LIBRARY ITEMS MY ITEMS WARNINGS				
Search by title				
by authors		by year		SEARCH
Title	Author	Year	Type	Status
Advances in artificial intelligence: applications and theory	Bezdek, James C.	2019	BOOK	HOLD ASSIGN TO STUDENTS
Kürk Mantolu Madonna	Bezdek, James C.	2019	BOOK	HOLD ASSIGN TO STUDENTS

In this page the instructor can see the library items and has the option to assign these items to their students via the “ASSIGN TO STUDENTS” button.

## 7.10. Register Library Item

**Bilkent Library**

LIBRARY ITEMS USERS REGISTER A NEW ITEM REGISTER A NEW USER

Register A Library Item

Select user type

BOOK JOURNAL

Title  
Advances in artificial intelligence: applicati

Publish Date  
2019-08-29

Language  
English

Description  
The book tells how to advances in artifica

Publisher  
World Scientific,

Authors  
Bezdek, James C.

Call Number  
VM-10

Edition  
20

Print Location  
Ankara

REGISTER

In this page, Librarian user can add a new item by first selecting the type from the top of the form: “BOOK” or “JOURNAL”. Then, they enter the item’s “Title”, “Publish Date”, “Language”, “Description”, “Publisher”, “Author”, “Call Number”, “Edition”, and “Print Location”. Finally, they select “REGISTER” to finalize the item registering process.

## 7.11. Warn User Page

**Bilkent Library**

LIBRARY ITEMS USERS REGISTER A NEW ITEM REGISTER A NEW USER

ALL STUDENT INSTRUCTOR

Search by id Search by last name SEARCH

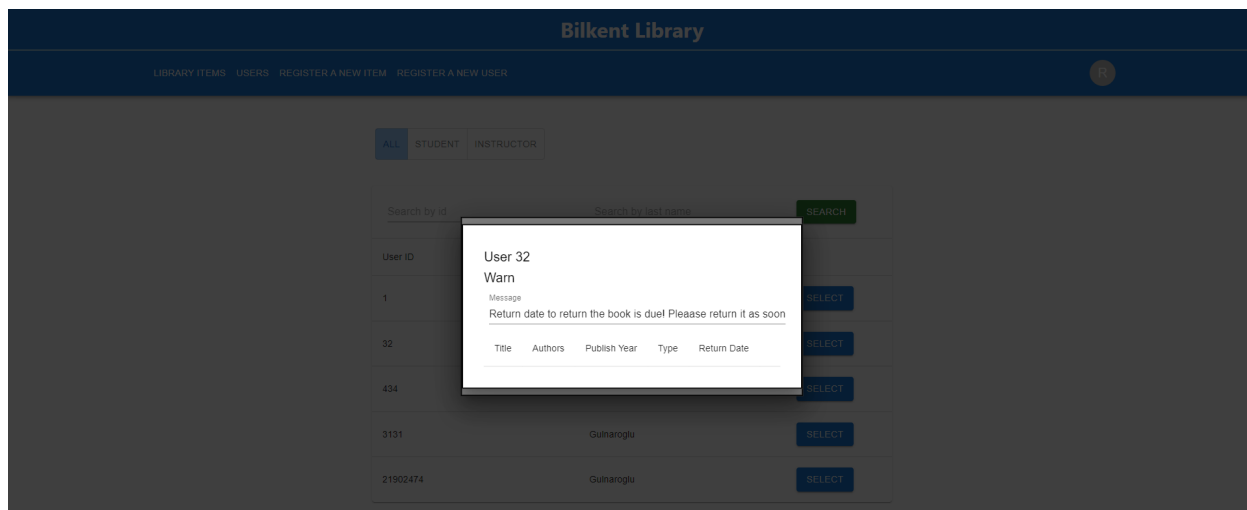
User ID	Last Name	SELECT
1	Cnel	SELECT
32		SELECT
434		SELECT
1234		SELECT
3131		SELECT
21902474	Guhanoglu	SELECT

User 32

View Items ON-HOLD BORROWED RETURNED

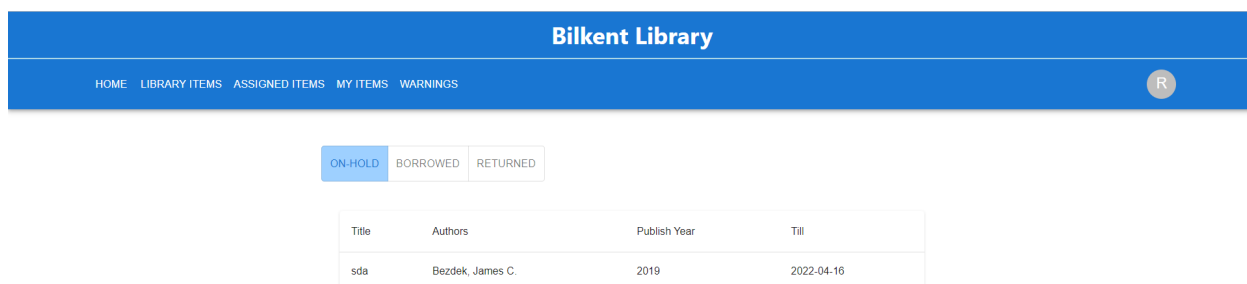
Actions LEND ITEM RETURN ITEM WARN

To warn a user, Librarian first needs to select the user from the main page then a pop-up section will show up, where they can select “WARN” option.



After the Librarian select the “WARN” option this popup will show where Librarian enters the warning message for the specific user and can customize it accordingly.

## 7.12. View My Items Page



User can view their borrowed, on hold, or returned items from this page by clicking the shown buttons.

## 7.13. View Warnings



**No Warnings**

The warnings a user has received will appear on this page.

## 7.14. Add Course Page



Course ID	Course name	
1	CS353	<a href="#">ADD COURSE</a>
2	CS319	<a href="#">ADD COURSE</a>
3	CS223	<a href="#">ADD COURSE</a>
4	CS342	<a href="#">ADD COURSE</a>
5	ENG401	<a href="#">ADD COURSE</a>
6	MATH225	<a href="#">ADD COURSE</a>

Using this page, students will be able to enroll to existing courses and instructors will be able to select the courses they teach in order to be able to assign their students library items, making students who take that course able to receive these assignments.

## 8. Website

Project description website: <https://kaanozaltan.github.io/school-library-database>

## 9. References

[1] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database system concepts. New York, NY: McGraw-Hill, 2020.