



# CS 353

## Database Systems

### Spring 2021

### Final Report

#### Team Members

Ahmet Feyzi Halaç	21703026
Ege Şahin	21702300
Göktuğ Gürbüztürk	21702383
Zeynep Cankara	21703381

Section: 1

Group: 10

## 1. DESCRIPTION OF APPLICATION SYSTEM

Social Cataloging Platform for the Books is an application system that brings readers and authors together in one platform. The system provides many functionalities for different user types. Database system is used in order to keep and manipulate data about the users and other necessary information that are required to implement the functionalities in the system. In this platform there are three types of users which are author, librarian and user(readers).

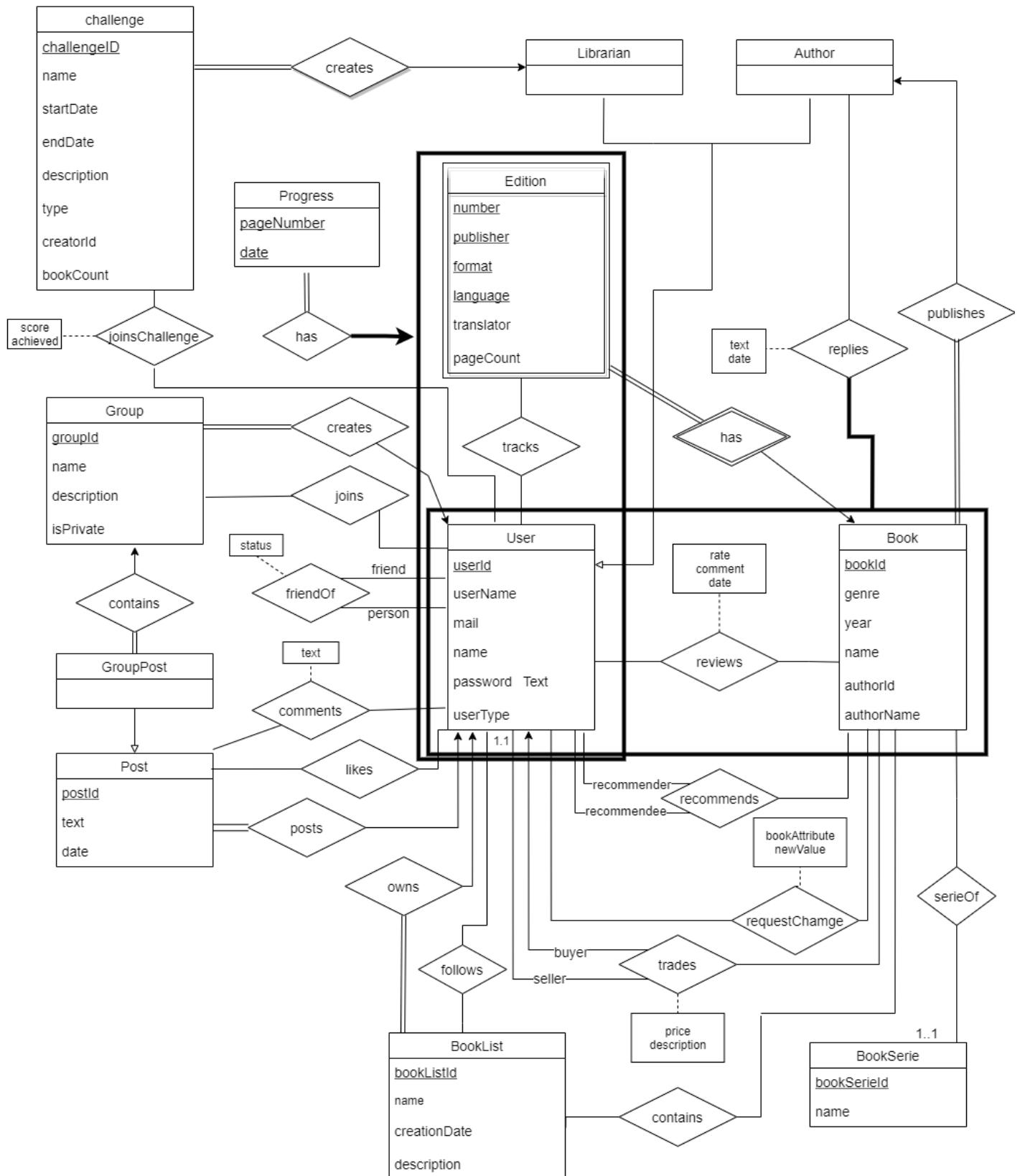
Standard users are able to see all the available books that are stored in the system via database. Also, they can start tracking the book and save their progress in the system. All the progress are visible in their profiles which provides them to know on which date and which page they stayed in the book. Additionally, they can create a booklist which can consist of several books and leave a review on the books in the system and create an erroneous info correction request. Users can join different kinds of challenges to encourage themselves to read more books. Lastly, as our additional functionality, we provide users to buy and sell books via this platform.

Authors are able to use the functionalities that are provided to standard users and some additional functionalities. The most important additional feature is publishing a book. They can specify the necessary information about the book and publish them in the system. Also, they can create editions of the published book. All these created books are stored in the database system and made visible to all other users. Finally, they can reply to reviews of her/his books, which is visible to all other users.

The last user type in the system is librarian. Librarians are responsible for two main things. The first one is listing all the request changes that are created by the authors or users and correct them if necessary. The second functionality is creating reading challenges for specified time intervals to make reading more fun for all the users and authors.

All these functionalities given above are provided by the database management system consistently. Results of all the actions manipulates the database by the queries.

## 2. E/R DIAGRAM



### 3. LIST OF TABLE SCHEMAS

User(userId, userName, name, mail, password, userType)

Book(bookId, genre, year, name, authorId, authorName)

Foreign Key: authorId references User (userId)

authorName references User(name)

Challenge(challengeId, name, startDate, endDate, description, type, creatorId, bookCount)

Foreign Key: creatorId references User (userId)

JoinsChallenge(challengeId, userId, score, achieved)

Foreign Key: challengeId references Challenge(challengeId)

userId references User(userId)

FriendOf(friendId, personId, status)

Foreign key: friendId references User(userId)

personId references User(userId)

Post(postId, text, date, writerId)

Foreign Key: writerId references User(userId)

Likes(postId, userId)

Foreign Key: postId references Post(postId)

userId references User(userId)

Comments(postId, userId, text)

Foreign Key: postId references Post(postId)

userId references User(userId)

BookList(bookListId, name, creationDate, description, ownerId)

Foreign Key: ownerId references User(userId)

Follows(userId, bookListId)

Foreign Key: userId references User(userId)

bookListId references BookList(bookListId)

Contains(bookListId, bookId)

Foreign Key: bookListId references BookList(bookListId)

bookId references Book(bookId)

Edition(bookId, number, publisher, format, language, pageCount, translator)

Foreign Key: bookId references Book(bookId)

Tracks(userId, bookId, number, publisher, format, language)

Foreign Key: (bookId, number, publisher, format, language) references

Edition(bookId, number, publisher, format, language)

userId references User(userId)

Progress(pageNumber, date, userId, bookId, number, publisher, format, language)

Foreign Key: (userId, bookId, number, publisher, format, language)  
references Tracks (userId, bookId, number, publisher,  
format, language)

Reviews(userId, bookId, rate, comment, date)

Foreign Key: userId references User(userId)  
bookId references Book(bookId)

Recommends(recommendeId, recommenderId, bookId)

Foreign Key: recommendeId references User(userId)  
recommenderId references User(userId)  
bookId references Book(bookId)

RequestChange(userId, bookId, bookAttribute, newValue)

Foreign Key: userId references User(userId)  
bookId references Book(bookId)

BookSerie(bookSerield, name)

SeriesOf(bookId, bookSerield)

Foreign Key: bookId references Book(bookId)  
bookSerield references BookSerie(bookSerield)

Replies(userId, bookId, authorId, date, text)

Foreign Key: authorId references User(userId)  
(bookId, userId) references Reviews(bookId, userId)

Trades(offerId, buyerId, sellerId, price, description, bookId)

Foreign Key: buyerId references User(buyerId)  
sellerId references User(userID)  
bookId references Book(bookId)

Grup(groupId, name, description, isPrivate, userId)

Foreign Key: userId references User(userId)

JoinsGroup(groupId, userId)

Foreign Key: groupId references Grup(groupId)  
userId references User(userID)

GroupPost(postId, groupId)

Foreign Key: postId references Post(postId)  
groupId references Grup(groupId)

## 4. SQL Queries of Common Functionalities

### Common Functionality 1

- **Login Validation**

```
SELECT *
FROM User
WHERE username = nameInput AND password = passwordInput
```

- **Create New User:**

- **Get all IDs to create new unique ID:**

```
SELECT userId
FROM User
```

- **Create user:**

```
INSERT INTO User
VALUES (newId, usernameInput, mailInput, nameInput, passwordInput,
specifiedUserType)
```

### Common Functionality-2 (Additional Functionalities)

- **List all offers for a book:**

```
SELECT *
FROM Trades
WHERE bookId = specifiedBooksId AND buyerId IS NULL
```

- **Buy a book:**

```
UPDATE Trades
SET buyerId = currentUsersId
WHERE offerId = specifiedOfferId
```

- **Sell a book:**

```
INSERT INTO Trades
VALUES (newId, NULL, currentUsersId, specifiedPrice, description, specifiedBookId)
```

- **Create group:**

```
INSERT INTO Group
VALUES (newId, specifiedName, description, isPrivate, currentUsersId)
```

- **Search groups:**

```
SELECT *
FROM Group
WHERE name LIKE '%searched_name%'
```

- **Join a group:**

```
INSERT INTO JoinsGroup
VALUES (specifiedGroupsId, currentUsersId)
```

- **Create new post in a group:**

```
INSERT INTO Post
VALUES (newID, specifiedText, currentDate)
```

```
INSERT INTO GroupPost
VALUES (newID, specifiedGroupId)
```

- **List all posts of a group:**

```
SELECT text, date
FROM (SELECT postId
      FROM GroupPost
      WHERE groupId = specifiedGroupId)
NATURAL JOIN Post
```

## SQL Queries For Social Cataloging Platform for Books

### Track a book progress (by a standard user)

- **List all available books and apply filters (genre, author, keyword):**

```
SELECT *
FROM Book
WHERE genre like '%specified keyword%' or author like '%specified keyword%'
```

- **Get all editions of a book:**

```
SELECT * FROM Book NATURAL JOIN Edition WHERE bookId = specifiedId
```

- **Select a book (and the edition) and start tracking:**

```
INSERT INTO Tracks
VALUES (userId, bookId, number, publisher, format, language)
```

- **Add progress to a book track:**

```
INSERT INTO Progress
```

VALUES (specifiedpageNumber, specifieddate, specifieduserId, specifiedbookId, specifiednumber, specifiedpublisher, specifiedformat, specifiedlanguage)

- **Get all progress information for a track:**

```
SELECT pageNumber, date  
FROM Progress  
WHERE (pageNumber, date, userId, bookId, number, publisher, format, language) =  
(specifiedpageNumber, specifieddate, specifieduserId, specifiedbookId,  
specifiednumber, specifiedpublisher, specifiedformat, specifiedlanguage)
```

- **Get all booklist IDs to create new unique ID:**

```
SELECT bookListId  
FROM BookList
```

- **Create book list:**

```
INSERT INTO BookList  
VALUES (newID, specifiedName, currentDate, description)
```

- **Add book to book-list:**

```
INSERT INTO Contains  
VALUES (specifiedBookListId, bookId)
```

- **List all users:**

```
SELECT *  
FROM User
```

- **Send friend request:**

```
INSERT INTO FriendOf  
VALUES (selectedUsersId, currentUserId, 'PENDING')
```

- **List all friend requests**

```
SELECT *  
FROM FriendOf JOIN User ON friendId = userId  
WHERE personId = specifiedUserId AND status = 'PENDING'
```

- **Accept friend request:**

```
UPDATE FriendOf SET status = 'ACCEPTED'  
WHERE personId = currentUserId AND friendId = specifiedFriendsId
```

- **Like their posts:**

```
INSERT INTO Likes  
VALUES (specifiedPostId, currentUserId)
```

- **Comment on their posts:**

```
INSERT INTO Comments  
VALUES (specifiedPostId, currentUserId, comment)
```

- **Get when specified book is read by user:**

```
SELECT date  
FROM Tracks NATURAL JOIN Progress  
WHERE userId = currentUserId AND bookId = specifiedBookId AND pageCount =  
pageProgress
```

- **Recommend a book to friends:**

```
INSERT INTO Recommends  
VALUES (recommendedUserId, currentUserId, bookId)
```

### **Publish Book (by an author)**

- **Specify features like title, description etc.**

```
INSERT INTO Book VALUES  
(bookId, genre, year, name, authorId, authorName )
```

- **Create several editions of the book**

- **Select the book to create its editions**

```
SELECT bookId  
FROM Book  
WHERE bookId = specifiedId
```

- **Specify edition format, page count, language for each edition**

```
INSERT INTO Edition VALUES  
(bookId, number, publisher, pageCount, format, language, translator)
```

- **Enter translator info if necessary**

```
UPDATE Edition  
SET translator = specifiedTranslator  
WHERE (bookId, number, publisher, pageCount, format, language) =  
(specifiedBookId, specifiedNumber, specifiedPublisher, specifiedPageCount,  
specifiedFormat, specifiedLanguage)
```

- **Review a book (by a standard user)**

- **Users selects a book and leave a review**

```
INSERT INTO Reviews VALUES
((SELECT bookId
FROM Book
WHERE bookId = specifiedId), userId, rate, comment, date)
```

- **Author select his/her book and list all available reviews**

```
SELECT *
FROM Reviews
WHERE bookId = specifiedBookId
```

- **Author selects a review and replies to it.**

```
INSERT INTO Replies VALUE
(selectedUserId, selectedBookId, currentDate, text, authorId)
```

### **Site Management (by a librarian)**

- **Erroneous info correction request (by an author or by a standard user)**
  - **User/Author select a book and creates a request**

```
INSERT INTO RequestChange VALUES
(userId, selectedBookId, bookAttribute, newValue)
```

- **Librarian lists all available requests**

```
SELECT *
FROM RequestChange
```

- **Select a request and approve/reject it**

```
DELETE FROM RequestChange
WHERE (userId, bookId, bookAttribute, newValue)
      = specifiedUserId, specifiedBookId,
      specifiedBookAttribute, specifiednewValue)
```

- **If approve, edit the information about book accordingly**

```
UPDATE Book
SET bookAttribute = newValue
WHERE bookId = specifiedBookId
```

- **Reading challenge**
  - **Specify title, deadline, type, book count etc.**

```
INSERT INTO Challenge
```

VALUES (specifiedChallengeId, specifiedName, specifiedStartDate,  
specifiedEndDate, specifiedDescription, specifiedType, creatorId)

- **List all available challenges and join one (by a standard user)**

```
SELECT *
FROM Challenges
INSERT INTO JoinsChallenge VALUES
(selectedChallengeId, userId, score)
```

- **See all participants and their progress of challenge**

```
SELECT userId, score
FROM JoinsChallenge NATURAL JOIN User
WHERE challengeId = selectedChallengeId
```

- **Outcome of challenge**

1. **If the user is successful, show it in the user's profile**

```
SELECT name, achieved
FROM JoinsChallenge NATURAL JOIN Challenge
WHERE ((userId = specifiedUserId) AND (achieved = 1))
```

2. **If the user fails, notify the user of failure.**

```
SELECT name, achieved
FROM JoinsChallenge NATURAL JOIN Challenge
WHERE ((userId = specifiedUserId) AND (achieved = 0))
```

## 5. Advanced DB Features

- **Views**

```
CREATE VIEW all_challenges AS
    SELECT *
        FROM challenges;
```

```
CREATE VIEW active_challenges AS
    SELECT *
        FROM challenges
        WHERE endDate - currentDate > 0;
```

```
CREATE VIEW championships AS
    SELECT *
        FROM Challenge NATURAL JOIN JoinsChallenge
        WHERE winnerId = userId
```

These views are used based on the user types. If the usertype is librarian then he/she gets all the challenges in the system. If the account belongs to a standard user then only available challenges become visible for him/her.

- **Range and Flexible Conditions**

```
SELECT *
    FROM Book
    WHERE genre LIKE '%specified keyword%' or
          authorName LIKE '%specified keyword%' or
          name LIKE '%specified keyword%'
```

**Output (specified keyword = ann):**

bookId	genre	year	name	authorId	authorName
2	fiction	2020	Dear Edward	4	Ann Napolitano
10	fantasy	2020	Piranesi	12	Susanna Clarke

```
SELECT *
    FROM Book
    WHERE year BETWEEN year1 AND year2
```

**Output (year1 = 2000, year2 = 2010):**

bookId	genre	year	name	authorId	authorName
13	fantasy	2003	Harry Potter and the Sorcerers Stone	15	J.K. Rowling
15	fantasy	2004	Harry Potter and the Prisoner of Azkaban	15	J.K. Rowling

• **Reports**

**The Most Popular 10 Challenges:**

This report is used in the main page to show 10 most popular challenges according to user preferences in descending order.

```
SELECT name, totalParticipant
FROM (SELECT name, challengeId, totalParticipant
      FROM Challenge
      NATURAL JOIN
      (SELECT challengeId, COUNT(score) AS totalParticipant
       FROM JoinsChallenge
       GROUP BY challengeId) AS temp) AS temp2
ORDER BY totalParticipant DESC
LIMIT 10
```

**Output:**

Name	totalParticipation
Challenge2	9
Challenge3	3
Challenge4	3
Challenge10	3
Challenge1	2
Challenge5	2
Challenge6	1
Challenge7	1
Challenge8	1
Challenge9	1

### **The most popular 10 book:**

This report is used in the main page to show 10 most popular books according to user preferences in descending order.

```
SELECT bookId, COUNT(userID) AS totalTrack, authorName, genre, name
FROM
  (SELECT userId, Tracks.bookId, temp.authorName, temp.genre, temp.name
   FROM Tracks
   JOIN
     (SELECT bookId, authorName, genre, name, year
      FROM Book) AS temp
    ON Tracks.bookId = temp.bookId) AS temp2
GROUP BY bookId, authorName, genre, name
ORDER BY totalTrack desc
LIMIT 10
```

### **Output:**

bookId	totalTrack	authorName	genre	name
1	6	Jennifer Weiner	fiction	Big Summer
8	5	Brit Bennett	historical	The Vanishing Half
3	2	Charlotte McConaghy	fiction	Migrations
5	2	Jess Lourey	mystery	Unspeakable Things
7	2	Alice Hoffman	historical	Magic Lessons
10	2	Susanna Clarke	fantasy	Piranesi
12	2	Patricia Briggs	fantasy	Smoke Bitten
2	1	Ann Napolitano	fiction	Dear Edward
4	1	Tarryn Fisher	mystery	The Wives
6	1	Adam Lazarre	mystery	Blacktop Wasteland

### **The Most Tracked Book:**

This report is used in the main page to show the most popular book according to user preferences.

```
SELECT temp3.bookId, MAX(totalTrack) AS maxTrack, authorName, genre, name
FROM
(SELECT bookId, COUNT(userID) AS totalTrack, authorName, genre, name
FROM
(SELECT userId, Tracks.bookId, temp.authorName, temp.genre, temp.name
FROM Tracks
JOIN
(SELECT bookId, authorName, genre, name, year
FROM Book) AS temp
ON Tracks.bookId = temp.bookId) AS temp2
GROUP BY bookId, authorName, genre, name) as temp3
```

bookId	maxTrack	authorName	genre	name
1	6	Jennifer Weiner	fiction	Big Summer

- **Trigger**

### **Check date of challenge:**

This trigger checks new insertions to challenge relation. If the start date of the challenge is later than the end date it deletes the new added challenge.

```
CREATE TRIGGER challengeDateChk AFTER INSERT ON challenge
FOR EACH ROW
BEGIN
IF NEW.startDate > NEW.endDate THEN
    DELETE FROM challenge WHERE challengeId = NEW.challengeId;
END IF;
END;
```

### **Update Score of Challengers:**

This trigger checks new insertions on progress relation. If the newly added progress has a page number equal to the book's page number and if the user is participating in a challenge, the user's score for the challenge is incremented.

```
CREATE TRIGGER updateChallengeScore AFTER INSERT ON progress
FOR EACH ROW
BEGIN
IF NEW.pageNumber = (SELECT pageCount
FROM Edition
```

```

        WHERE NEW.bookId = Edition.bookId AND
        NEW.number = edition.number)
    AND NEW.date < (SELECT challenge.endDate
                    FROM challenge NATURAL JOIN joinschallenge
                    WHERE NEW.userId = joinschallenge.userId)
    AND NEW.date > (SELECT challenge.startDate
                    FROM challenge NATURAL JOIN joinschallenge
                    WHERE NEW.userId = joinschallenge.userId)
THEN
    UPDATE joinschallenge
    SET score = score + 1
    WHERE NEW.userId = joinschallenge.userId;
END IF;
END;

```

- **Procedure**

**STORED PROCEDURE: Getting challenge and success pairs for specified user**

```
CREATE PROCEDURE getAchievementInfo(IN specifiedUserId INT)
```

```
BEGIN
```

```
    SELECT name,achieved
    FROM JoinsChallenge NATURAL JOIN Challenge
    WHERE ((userId =specifiedUserId) AND (achieved IS NOT NULL));

```

```
END
```

This procedure is used to get the success information on challenges of a specific user in order to show the achievement and failure status on the user's profile.

**STORED PROCEDURE: Indicate the achievements of users when the challenge is finished**

```
CREATE PROCEDURE indicateAchievement(IN specifiedChallengeld INT)
```

```
BEGIN
```

```
    UPDATE JoinsChallenge
    SET achieved = 1
    WHERE (score >= (SELECT bookCount FROM Challenge WHERE challengeld =
specifiedChallengeld))
    AND
    (challengeld = specifiedChallengeld);

```

```
    UPDATE JoinsChallenge
    SET achieved = 0
    WHERE (score < (SELECT bookCount FROM Challenge WHERE challengeld =
specifiedChallengeld))
    AND
    (challengeld = specifiedChallengeld);

```

```
END
```

Procedure given above is used to set the success or failure status of the participants of a specific challenge that is closed by the librarian. It works based on the goal and the score of the user at the end of the challenge.

- **Secondary indices**

The secondary indices provide efficient data access for the database other than the primary index. We created a secondary index on "genre" column of the Book relation for the "Searching book by genre" functionality

```
CREATE INDEX genreIndex ON Book (genre);
```

We created a secondary index on "bookCount" column of the Challenge relation for the "Searching challenge by book count" functionality

```
CREATE INDEX bookcountIndex ON Challenge (bookCount);
```

## 6. Implementation Details

For the Frontend of BilRead, our Social Book Cataloging Application, we used React [1]. It is an open-source, front end JavaScript library for building user interfaces or UI components [1]. For the Backend part of our Social Book Cataloging platform, we used Node.js[2]. It is an open-source JavaScript runtime environment built on Chrome's V8 JavaScript engine [2]. We decided to use Node.js because it enables building scalable, high performant, and high extendible applications quickly. Additionally, it provides non-blocking I/O systems which can handle the processing of simultaneous requests. We used MySQL for the persistence data storage and ran the database on our local machine. We chose MySQL precisely since we needed to run SQL queries according to the project requirements. Besides the project requirements, SQL databases are especially advantageous with the applications which have structured data format.

We divided the main work into sections as the front-end part which is responsible for the user interface, backend part for routing and middleware and the database part for the persistent data storing.

Following is the contribution of each member to the project:

### **Ahmet Feyzi Halaç**

Implementation of front-end except the Groups tab.

Initial construction of backend (initial connection to database, using express.js to construct template for endpoints)

### **Ege Şahin**

Re-construction of ER diagram. Worked on the backend for initializing and populating the database. Created two triggers and three reports. Also worked on asked queries and implementation of them to the code. Finally, wrote important parts of the report.

### **Göktuğ Gürbüztürk**

Mostly, worked on the backend. Implemented the endpoints and manipulated the database via the queries implemented inside them. Additionally, worked on initializing and populating the database. Also, created stored procedures, views and trigger and queries asked in the functionality document for both report and implementation. Finally, wrote some parts of the report.

### **Zeynep Cankara**

Writing the User Manual and Implementation details of the Final report. Worked on frontend for creating, joining group functionality. Created two secondary indices. Connected group function related queries to the frontend. Contributed in the report writing (Mockup UI designs).

## 7. User Manual

*BilRead* had three main types of users : the User, Author and the Librarian. In the User Manual we are giving an overview on the main functionality of our system in accordance with the project specifications document and the feedback we received for our design report.

### 7.0 Welcome Page

A new user who is not yet logged in to the platform is directed to our welcome page, which is our home page. The newcomers can then navigate within the website using the tabs on our navigation bar. There are tabs labelled as "Home", "Book", "Challenge", and "My Lists." On the right side of the navigation bar, the user can see their profile avatar, where the user can use it to log out if already signed in.

### 7.1 Common Functionality-1

#### 7.1.1 Sign-in/Login Page

To sign in for the *BilRead*, people should use the following page in Figure 1. As one can see from Figure 1. that a user needs to have the information of the user type, name/email and password associated with their accounts in *BilRead* to sign in properly. The people can select the user type of the account from the drill-down menu, and people can type their name/email and password credentials to the text fields provided to them. After hitting the "Sign In" button, the application will direct them to the "Home" page of the application. If they do not have an account, they can sign-up from the link below the password text field as indicated in Figure 1. As one can be seen from Figure 1. the unsuccessful attempts notify the user with a constructive error message why sign-in was unsuccessful.

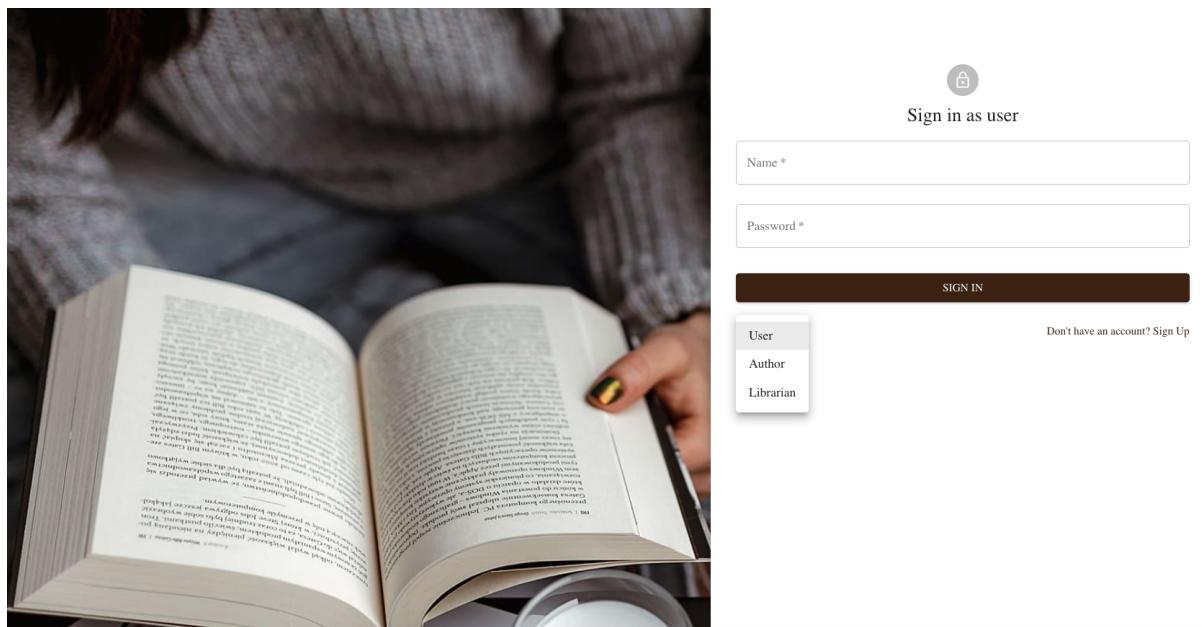


Figure 1. Sign-in as a user page

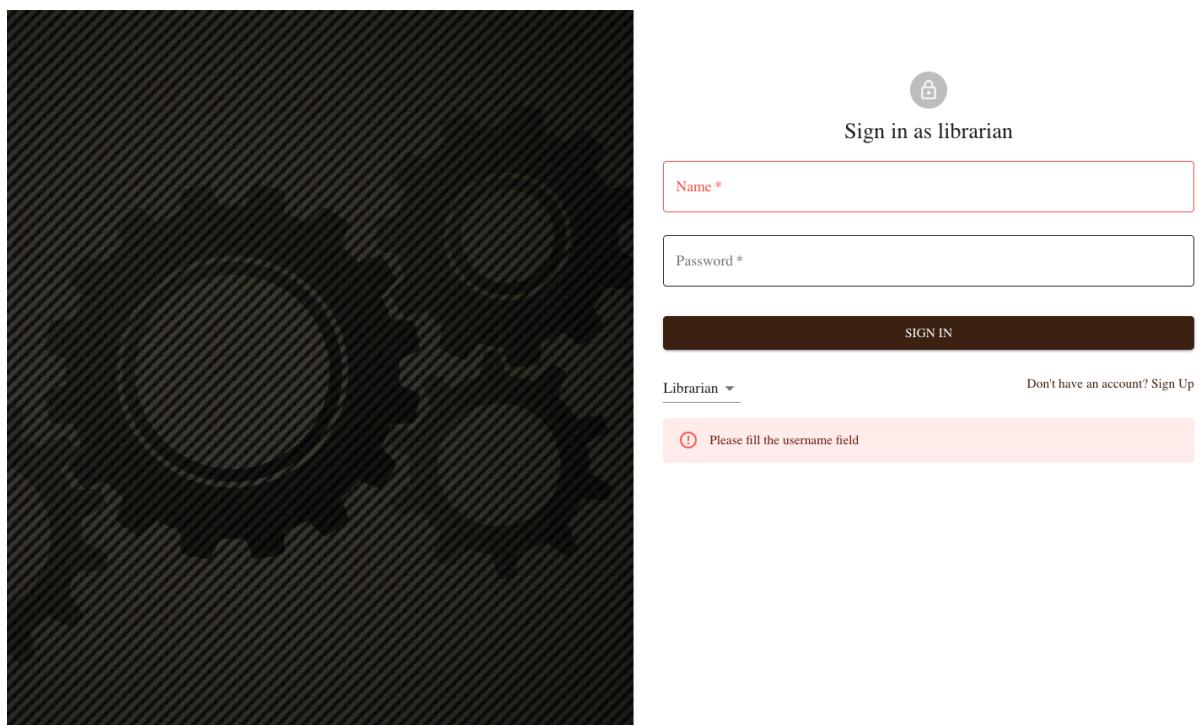


Figure 2. Unsuccessful Librarian Sign-in attempt

### 7.1.2 Sign-up Page

To sign up for the BilRead, people should use the following page in Figure 3. As one can see from Figure 3. that a user needs to type the information of the user type, name, username, email address and password, which they want to create an account for the BilRead. They need to use the drill-down to specify the user type and type the other credential information within the text fields provided for the sign-in page. After hitting the "Sign Up" button, the application will send the information to the database to properly sign in again.



🔒

**Sign up as author**

Name \*

Username \*

Email Address \*

Password \*

[SIGN UP](#)

Author ▾
Already have an account? [Sign in](#)

Figure 4. Sign-up as an Author page

### 7.1.3 Home Page

Following is the homepage of the "BilRead" social book cataloguing platform. The user will present this screen after they successfully signed in to the application. The leftmost pane displays the most popular ten challenges ordered by the number of participants. The middle pane displays the activity of the user's friends and the rightmost pane displays the most popular ten books in the system ordered by the number of people reading the book.

**Most Popular 10 Challenges**

<b>Challenge2</b>	13 participants
<b>Challenge10</b>	9 participants
<b>Challenge3</b>	3 participants
<b>Challenge4</b>	3 participants
<b>Challenge1</b>	2 participants
<b>Challenge5</b>	2 participants
<b>Challenge6</b>	1 participants
<b>Challenge7</b>	1 participants
<b>Challenge8</b>	1 participants
<b>Challenge9</b>	1 participants

**cconaghy reviewed Harry Potter and the Sorcerers Stone**

★★★★★ July 20th 2021

Ehh

---

**jlourey reviewed The Vanishing Half**

★★★★★ May 3rd 2021

I strongly recommend everyone to read it. It broadened my perspective.

---

**cconaghy reviewed Dear Edward**

★★★★★ April 20th 2021

There are contradictory statements in the book.

---

**egesah reviewed Big Summer**

★★★★★ April 10th 2021

It was a very gripping book.

---

**jlourey reviewed Harry Potter and the Sorcerers Stone**

★★★★★ April 9th 2021

Hate

---

**jkrowling reviewed Big Summer**

★★★★★ April 9th 2021

Selamun Alevkum

**Most Popular 10 Books**

<b>Big Summer</b>	by Jennifer Weiner
6 people reads this book	
<b>The Vanishing Half</b>	by Brit Bennett
5 people reads this book	
<b>Migrations</b>	by Charlotte McConaghy
2 people reads this book	
<b>Unspeakable Things</b>	by Jess Lourey
2 people reads this book	
<b>Magic Lessons</b>	by Alice Hoffman
2 people reads this book	
<b>Piranesi</b>	by Susanna Clarke
2 people reads this book	
<b>Smoke Bitten</b>	by Patricia Briggs
2 people reads this book	
<b>Dear Edward</b>	

Figure 5. Home page

#### 7.1.4 User Profile

The profile page of a user displays the successful challenges the user joined in. Additionally, the books the user bought from the platform will show up in the user profile.

The screenshot shows the Bilread user profile page for a user named 'egesah'. The top navigation bar includes links for Home, Books, Challenges, My Lists, and My Profile. The 'My Profile' link is highlighted. On the left, there's a sidebar with a circular profile picture containing the letter 'e' and the text 'Bought Books'. The main content area is divided into two sections: 'Challenge Outcomes' and 'Bought Books'. The 'Challenge Outcomes' section lists five challenges with their outcomes: Challenge1 (won), Challenge4 (won), Challenge6 (won), Challenge8 (lose), and Challenge9 (lose). The 'Bought Books' section is currently empty.

Figure 6. User Profile page

## 7.1.5 Books Page

From the "Books" tab users can see all the available books in the system.

The screenshot shows the Bilread Books page. At the top, there are navigation links: Home, Books (which is the active tab), Challenges, and My Lists. On the right, there is a user profile icon labeled 'egesah' with a small 'e' icon. Below the navigation, there is a search/filter section with fields for Book Name, Author, and Genre, and a Publish Year slider set between 1900 and 1981. Buttons for 'APPLY FILTER' and 'RESET FILTERS' are present. The main content area displays a grid of book cards. Each card includes the book's title, author, publication year, genre, and two buttons: 'SEE PROGRESS' and 'SEE REVIEW'. The books listed are: Big Summer, Migrations, Unspeakable Things, Magic Lessons, Deacon King Kong, Peace Talks, Dear Edward, The Wives, Blacktop Wasteland, The Vanishing Half, Piranesi, and Smoke Bitten. Each book card also has 'START READING' and 'REVIEW' buttons.

Figure 7. Listing all available books

From the "Books" tab users can filter books by various attributes such as name, author, genre, and publish year of the book.

The screenshot shows the Bilread Books page with applied filters. The search/filter section now has 'Magic Lessons' entered in the Book Name field. The Publish Year slider is set from 1900 to 2021. The main content area shows a single book card for 'Magic Lessons' by Alice Hoffman (2020) with genre historical. It features 'SEE PROGRESS' and 'SEE REVIEW' buttons, and 'START READING' and 'REVIEW' buttons below it. The rest of the page is mostly blank due to the applied filters.

Figure 8. Applying filters for book searching

The users can see the reviews done for a book by clicking on the "See Review" button from the "Books" tab.

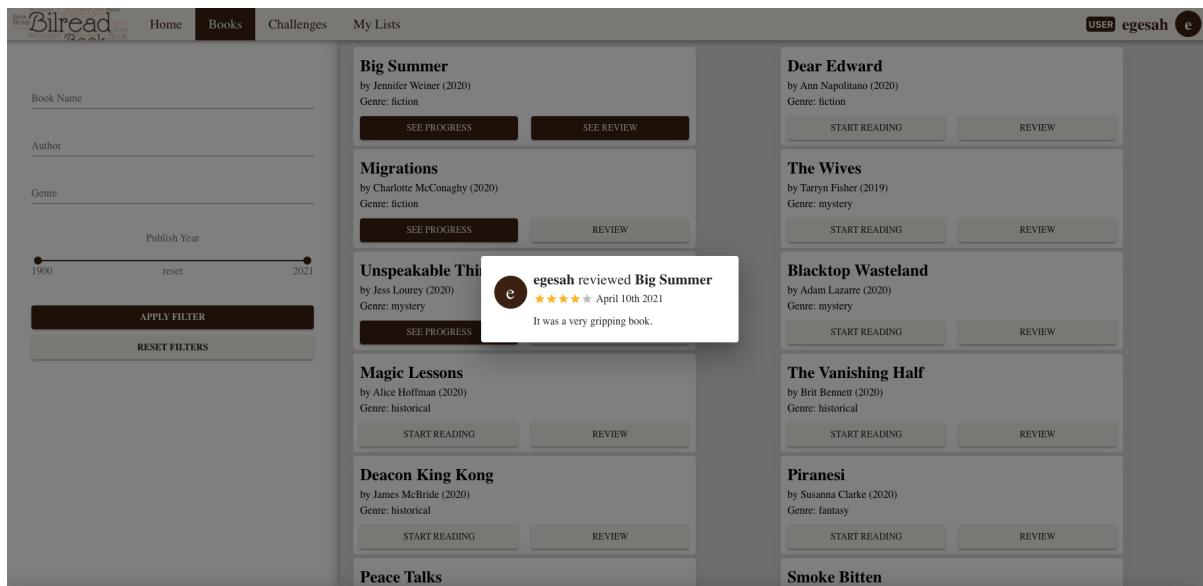


Figure 9. Seeing the reviews for a book

The users can use the "Books" tab to see available books on the platform. In order to start reading a book, they need to click on the "Start Reading" button. Then they will be presented with a pop up indicating which edition of the book they want to start reading.

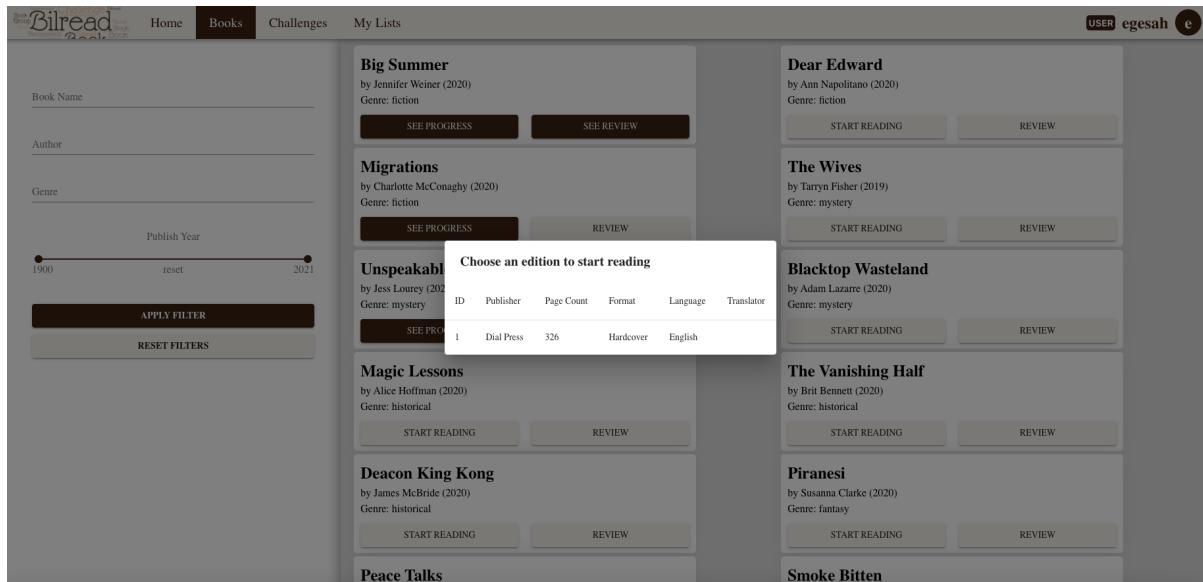


Figure 10. Starting to read a book by selecting the edition

Furthermore, users can drop reviews on the books by clicking on the "Review" button of the platform. A rating consists of a metric composed of five stars and a description. After clicking on the "Review" button the form will be submitted and the rating becomes visible in the system.

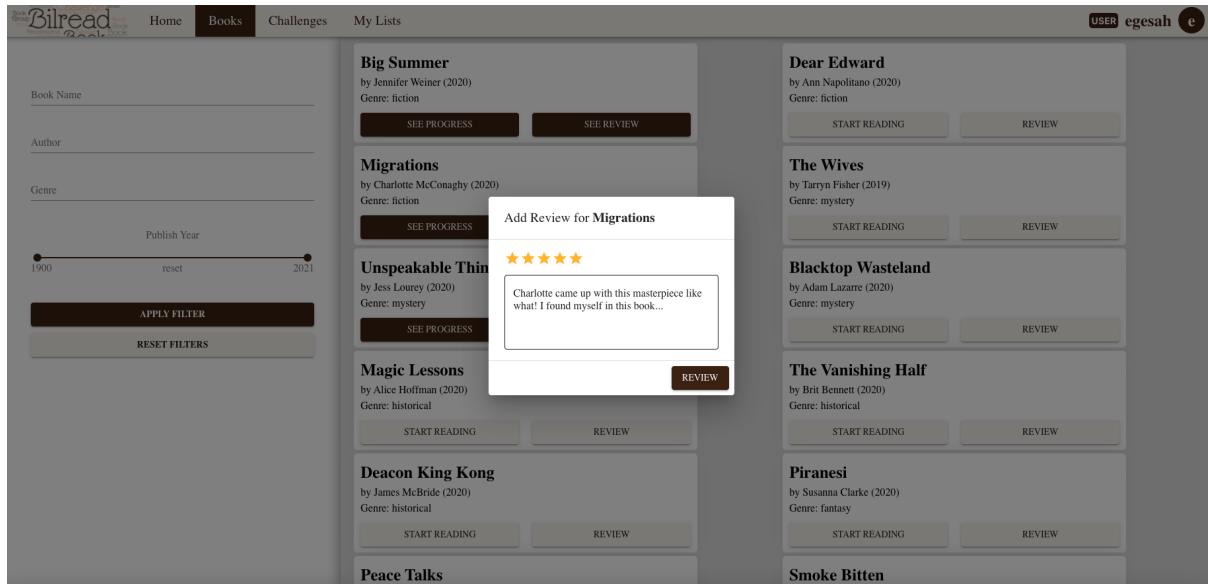


Figure 11. Rating a book

The users can track their book progress after clicking on the "See Progress" button from the "Book Page" of the application. The users can mark their progress with the number of pages they read and the date they want to put a bookmark on.

This screenshot shows the 'Migrations' book page. At the top, there are tabs for Home, Books, Challenges, My Lists, and Migrations (which is active). The book details are shown: 'Migrations' by Charlotte McConaghy (2020), genre: fiction. Below this is a table for 'EDITIONS' with two rows:

Edition Number	Publisher	Page Count	Format	Language	Translator
1	Flatiron Books	186	Hardcover	English	
2	Macmillan Audio	364	Paperback	English	

To the right, a 'Your Progress' section is displayed. It shows a vertical progress bar with markers for 'Page 10 in May 16th 2021' (green), 'Page 130 in May 16th 2021' (black), and 'End of book (186 page)' (grey). Below the bar is a form to add new progress: 'Page Number' (130) and 'Date' (16.05.2021). An 'ADD NEW PROGRESS' button is at the bottom.

Figure 12.1. Tracking book progress by the standard user

The screenshot shows a book detail page for "Unspeakable Things" by Jess Lourey (2020). The book is categorized under mystery. Below the title, there are tabs for "EDITIONS" and "OFFERS". A table provides details: Edition Number 1, Publisher Thomas & Mercer, Page Count 264, Format Hardcover, Language English. To the right, a "Your Progress" section shows a green vertical bar reaching the 264-page mark, with a note: "Page 264 in May 16th 2021". A message box says: "You finished reading this book. Congratulations!"

Figure 12.2. Tracking book progress by the standard user

### 7.1.6 Books Lists Page

The users can create their own book lists from the "My Lists" tab of the application. They can manage their existing lists by adding and removing books from them. Additionally, they can create new book lists from the rightmost pane by providing a name and description for the new book list. Adding a book done by selecting the booklist to add the book into and then clicking on the "Add Books" button. They need to click on the available books to add to their book list. They can remove the books from their list by clicking on the bin icon next to the books in the book list.

The screenshot shows the "My Lists" page. On the left, there are two lists: "favorite" (Creation Date: January 1st 2021) and "random" (Creation Date: March 1st 2021). The main table lists books with columns: Book ID, Book Name, Genre, Author, and a delete icon. Books listed include Big Summer, Dear Edward, Migrations, The Wives, Unspeakable Things, Magic Lessons, Deacon King Kong, Peace Talks, and Smoke Ritten. At the bottom is an "ADD BOOKS" button. On the right, a sidebar titled "Create New List" has fields for "Booklist Name" and "Description", and a "CREATE" button.

Figure 13. Creating booklists

The screenshot shows the Bilread platform's user interface. At the top, there are navigation links: Home, Books, Challenges, My Lists, and a user profile icon for 'egesah'. The 'My Lists' tab is active. On the left, there's a sidebar with 'My Lists' sections: 'favorite' (Creation Date: January 1st 2021) and 'random' (Creation Date: March 1st 2021). The main content area displays a table of books. A modal window titled 'Select books you want to add to favorite' is open, listing 12 books from a larger collection. Each book entry includes its ID, name, genre, author, and a small red square icon. At the bottom right of the modal is a large 'ADD' button.

Figure 14. Adding books to the book lists

The users can use the "Challenges" tab to see available challenges in the platform. By clicking on a challenge they can see the participant list with their progress. They can join a challenge by clicking on the "Join" button.

The screenshot shows the Challenges page of the Bilread platform. At the top, there are navigation links: Home, Books, Challenges, My Lists, and a user profile icon for 'egesah'. The 'Challenges' tab is active. The main content area displays a table of challenges. Each challenge entry includes its title, start date, end date, description, type, and book count, followed by a 'JOIN' button. To the right of the challenges table is another table titled 'Participants' which lists the names and currently read book counts of the users.

Title	Start Date	End Date	Description	Type	Book Count	JOIN
Challenge2	February 2nd 2021	July 2nd 2021	description2	reading	40	<button>JOIN</button>
Challenge3	February 1st 2021	August 2nd 2021	summer challenge	reading	50	<button>JOIN</button>
Challenge5	February 3rd 2021	August 2nd 2021	summer challenge3	reading	50	<button>JOIN</button>
Challenge7	February 5th 2021	August 2nd 2021	medium level	reading	80	<button>JOIN</button>
Challenge10	February 8th 2021	August 2nd 2021	hard level	reading	130	<button>JOIN</button>

Participants	Currently Read Book Count
Jim Butcher	25
James McBride	20
Brit Bennett	19
Alice Hoffman	18
Adam Lazarre	17
Jess Lourey	16
Tarryn Fisher	15
Charlotte McConaghy	14
Ann Napolitano	13
Jennifer Weiner	12
Susanna Clarke	11
goktug gurbuzturk	10
ege sahin	5

Figure 15. Challenges page

Screens for the additional functionality of our system. The main additional functionality going beyond the project functionalities was the use of the platform for selling and buying books offered by the users.

The screenshot shows a user interface for a book platform. At the top, there is a navigation bar with links: Home, Books, Challenges, My Lists, and Big Summer (which is highlighted). On the right side of the top bar, there is a user profile icon labeled 'egesah' with a small 'e' icon.

The main content area displays a book titled 'Big Summer' by Jennifer Weiner (2020). It is categorized as fiction. Below the title, there are two sections: 'EDITIONS' and 'OFFERS'.

In the 'EDITIONS' section, there is a form titled 'Sell This Book' with fields for 'Price' and 'Description', and a 'SELL' button.

In the 'OFFERS' section, there is a listing for 'Second hand' from 'egesah'. It shows a price of '15\$' and a 'BUY' button.

To the right of the book details, there is a sidebar titled 'Your Progress' which lists reading milestones: 'Page 25 in February 5th 2021' and 'Page 50 in February 7th 2021'. Below this, it says 'End of book (254 page)' and has input fields for 'Page Number' (set to 0) and 'Date' (set to 16.05.2021), with a 'ADD NEW PROGRESS' button.

Figure 16. Creating bookselling requests and seeing the offers for buying the book

## 7.1.7 Authors My Books Page

The Authors can reply to the reviews done on their books from their profile under the "My Books" section. They need to click on the "See Reviews" button where they make the reply by typing in the text field and pressing the key "enter".

ID	Publisher	Page Count	Format	Language	Translator
1	Scholastic Inc	309	Hardcover	English	

ID	Publisher	Page Count	Format	Language	Translator
1	Pottermore Publishing	341	Hardcover	English	

ID	Publisher	Page Count	Format	Language	Translator
1	Scholastic Inc	435	Hardcover	English	

Figure 17. Reply to the reviews (by an author)

An author can create various editions of their books and see them getting listed on the "My Books" page of the author.

ID	Publisher	Page Count	Format	Language	Translator
1	Scholastic Inc	309	Hardcover	English	

ID	Publisher	Page Count	Format	Language	Translator
1	Pottermore Publishing	341	Hardcover	English	

ID	Publisher	Page Count	Format	Language	Translator
1	Scholastic Inc	435	Hardcover	English	

Figure 18. Creating several editions of the book (by an author)

The Authors can publish books from their profile under the "My Books" section. They need to click on the "Publish New Book" button, specify the related information click on the "Submit" button.

The screenshot shows the 'My Books' section of the Bilread platform. At the top, there's a navigation bar with 'Home', 'Books', 'Challenges', 'My Lists', and a user profile for 'jkrowling'. Below the navigation, there's a sub-navigation for 'My Books'.

**Publish New Book:** A form to add a new book. It includes fields for 'Book Title' (The Intelligent Investor), 'Genre' (Finance), and 'Publish Year' (2021). A 'PUBLISH' button is at the bottom.

**Harry Potter and the Sorcerers Stone:** A table showing the edition details. It has one entry: ID 1, Publisher Scholastic Inc, Page Count 309, Format Hardcover, Language English. Buttons for 'LIST REVIEWS' and 'ADD NEW EDITION' are present.

**Harry Potter and the Chamber of Secrets:** Another table showing the edition details. It has one entry: ID 1, Publisher Pottermore Publishing, Page Count 341, Format Hardcover, Language English. Buttons for 'LIST REVIEWS' and 'ADD NEW EDITION' are present.

**Harry Potter and the Prisoner of Azkaban:** A third table showing the edition details. It has one entry: ID 1, Publisher Scholastic Inc, Page Count 435, Format Hardcover, Language English. Buttons for 'LIST REVIEWS' and 'ADD NEW EDITION' are present.

Figure 19. Publishing a book (by an author)

### 7.1.8 Challenge Page

The Librarians can create challenges by selecting add a new challenge and pressing the "create" button.

The screenshot shows the 'Challenges' section of the Bilread platform. At the top, there's a navigation bar with 'Home', 'Books', 'Challenges', 'My Lists', and a user profile for 'ggurbuztuk'. Below the navigation, there's a sub-navigation for 'Challenges'.

**ADD NEW CHALLENGE:** A modal window titled 'Create New Challenge' with fields for 'Title', 'Start Date' (dd.mm.yyyy), 'End Date' (dd.mm.yyyy), 'Description', 'Type', and 'Book Count'. A 'CREATE' button is at the bottom right.

**Challenge List:** A table listing existing challenges. The columns are 'Title', 'Start Date', 'End Date', 'Type', and 'Book Count'. The challenges listed are: Challenge2, Challenge3, Challenge5, Challenge7, and Challenge10.

**Select Challenge to See Participants:** A sidebar on the right with five 'JOIN' buttons corresponding to the challenges listed in the main table.

Figure 20. Creating a challenge (by a librarian)

### 7.1.9 Group Page

The users can create and join groups from the Groups tab by selecting add a new group button and then pressing the "create" button. They can also join to the already created groups.

The screenshot shows the 'Groups' section of the Bilread Book Club website. At the top, there's a navigation bar with links for Home, Books, Challenges, My Lists, and Groups. On the right, there's a user profile for 'egesah' with a dark circular icon. Below the navigation, there's a table listing existing groups:

Name	Description	Action
Stars	Group of stars	JOIN
Readers	Group of book lovers	JOIN
Bookworms		
Readers on the storm		

A modal window titled 'Create New Group' is open in the center. It contains two input fields: 'Title' with the value 'Readers on the storm' and 'Description' with the value 'Just another group'. A 'CREATE' button is at the bottom of the modal.

On the right side of the screen, there's a sidebar titled 'Participants' showing two entries: 'ege sahin' and 'Jennifer Weiner'.

Figure 21. Creating a group / see participants (by a user)

## 8. References

- [1] "React – A JavaScript library for building user interfaces", ReactJS. [Online]. Available: <https://reactjs.org/>. [Accessed: 08-Feb-2021].
- [2] "Node.js", Node.js. [Online]. Available: <https://nodejs.org/>. [Accessed: 08-Feb-2021].
- [3] "Node.js web application framework", ExpressJS. [Online]. Available: <https://expressjs.com/>. [Accessed: 08-Feb-2021].