CS 353 Database Systems

Spring 2021

Project Proposal

Team Members

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

Section: 1

Group: 10

Instructor: Uğur Güdükbay

Contents

1 Introduction

This report is a proposal for the "Social Cataloging Platform for the Books" project. In the proposal, we mentioned why and how a database is used as a system to provide functionality of a social book cataloging platform like "goodreads" with additional social features we implemented for the project such as integrating book selling and buying options and forming social groups which help people with similar book interests to come together. Additionally, we provided the requirements of the project under two main subtitles which are functional and nonfunctional requirements. The functional requirements are the functions provided by the system, and they define the contract of how the system should behave with the given inputs. On the other hand, non-functional requirements focus on the user's expectations of how the system should behave and they specify the metrics to measure the service level objectives in order to specify how the system will operate functionally and how it will handle incidents. We listed some constraints and boundaries of the application in the limitations section. Finally, we provided the conceptual design of the database using the E / R model for the "Social Cataloging Platform for the Books" project.

2 Project Description

As a Database Systems course requirement, the project assigned to our group is implementing "Social Cataloging Platform for the Books". The system will be designed for supporting necessary use cases for the users, authors and librarians. The system will allow users to track, like, comment, and rate the books on the system. The users can add friends, and see their friends’ book lists. The users will also be able to participate in challenges created by the librarians. Additionally, they can create posts and see posts of their friends' posts in their feed in order to utilise social features of the book cataloging platform. The users can be an active participant of the community by being able to make modifications on the book information by contacting the librarians. The authors will be able to publish books and engage with their readers by being able to reply to user comments. Finally, the librarians will act as a system admin and create challenges and solve the request for making modifications on the books in the system. We are planning to use React for the front-end and NodeJS for the back-end. React allows creating dynamic UI components easily and compatible with the NodeJS back-end. For our database we are planning to use MySQL which will allow us to send SQL queries and create, retrieve, update data.

2.1 Reasons to Use a Database System

The Social Book Cataloging Platform contains large amounts of data with variety such as users, authors and books. The actors of the system can perform various actions, for the users of the system for example it is creating book lists, participating in book challenges, publishing posts and tracking their progress on the books they read. The use of a database is necessary to store large volumes of data. A database system allows us to manage data easily by forming complex queries to make modifications on the existing data. The data is accessible and persistent with the use of a database system. Database systems can configure the access permissions to provide different levels of authorization to certain parts of the data stored in the database. Additionally, databases can backup the data in case a certain crash or corruption happens which makes them a reliable option to store the data. That's why we use databases as a way of storing and accessing the interrelated data easily and efficiently in our application.

2.2 How The Database Going to Be Used

Databases allow us to store data with relations in a persistent manner. Since the data we will use has many relations within we need a way to fetch data by forming complex queries. Using the database, we will be able to query relations and entities to accomplish this goal. Initially we will create new data entries in the system and make necessary updates to perform various actions such as when the user creates a book list, an author publishes a book, or the librarian creates challenges.

3 Requirements

3.1 Functional Requirements

3.1.1 User

Users can track books they have read. Users should be able to search for books by author, genre, year etc., mark their progress on books, rate and comment on books, create book lists, add friends and like/comment on their posts and recommend books to their friends

3.1.1.1 Search a book

Users can search a book by author, genre, year.

3.1.1.2 Mark progress on book

Users can mark the progress on a book. So, when he/she logs back into the system with his/her account, he/she can continue reading from where he/she left off.

3.1.1.3 Add friends

Users has an option to add friends from the username or user-id identifiers that are unique to each user.

3.1.1.4 Rate a book

Users can rate a book with numbers from 1 to 10, so that the other users can look at the rating of the book when he/she tries to decide whether to read or not.

3.1.1.5 Comment on a book

Users can comment on a book via the provided text field. The comment will be visible to the other users. So these comments will give an idea to other people who will read the book.

3.1.1.7 Create a book list

Each user can create a book list consisting of books chosen by the person(creator). Also, the user can share the list with his/her friends.

3.1.1.8 Post

Users can post something on their profiles. And their friends should be able to see these posts.

3.1.1.9 Comment on a post

Users can make a comment on their friends’ posts via the provided text field. The comments will be visible to other users who are able to see the post.

3.1.1.10 Like a post

Users can like their friends’ posts.

3.1.1.11 Follow a book list

Users can follow the book lists of their friends.

3.1.1.12 Recommend a book

Users can recommend a book to their friends.

3.1.1.13 Joins challenge

Users can participate in challenges which are created by the Librarian.

3.1.1.14 Request a change

Users can request a change for a specified book. When he/she thinks that the attribute of the book is incorrect and it should be changed, he/she will be able to request a change to correct this error.

3.1.1.15 Buy/Sell a book

Users can act as both buyers and sellers on this platform. Users can buy books that will be sold by other users. Offers will be shown on the page of the book.

3.1.1.16 Group

There will be groups on the system for specific topics which are created by the user. Other users can search these groups in the system like searching for a book, join those groups and post to those groups. If the group is not private, these posts can also be seen by other users who are not members of that group. Otherwise group posts can only be seen by the members.

3.1.2 Librarian

3.1.2.1 Create a reading challenge

Librarians can create reading challenges which users can join.

3.1.2.2 Confirm the change request

Librarians can confirm users’ or authors’ change requests and edit erroneous information about books.

3.1.2.2 Decline the change request

Librarians can decline a change request.

3.1.3 Author

3.1.3.1 Publish a Book

Authors can publish their new books on the system.

3.1.3.2 Reply to a review

Authors can view the reviews on their book and reply to those reviews if they want.

3.1.3.3 Request a change

Authors can request a change for a specified book. When he/she thinks that the attribute of the book is incorrect and it should be changed, he/she will be able to request a change to correct this error.

3.2 Non-functional Requirements

3.2.1 Authentication & Security

System will use email verification to authenticate users so that there cannot be any vulnerability. Also, passwords of the users will be hashed before inserting to corresponding tables in the database. This prevents unwanted access to users' passwords.

3.2.2 User-Friendliness

User interface of the website will be easy-to-use, where button numbers are minimal so that there will be no confusion about the actions users can take in any page. Users can perform their requests in a few steps, which will reduce the time spent on all actions.

3.2.3 Reliability

System will be implemented in a way that even if there are errors on a page, it will not affect the overall user experience, users can still interact with the page without any problem. Also, the system will backup data regularly so that in case of a serious crash, all data about the system will not be lost.

3.2.4 Capacity

System will accomodate large number of users simultaneously without any problem.

3.2.5 Efficiency

System will make effective database queries. Also, results of the frequently accessed queries can be stored as well to reduce the query number to database.

3.2.6 Portability

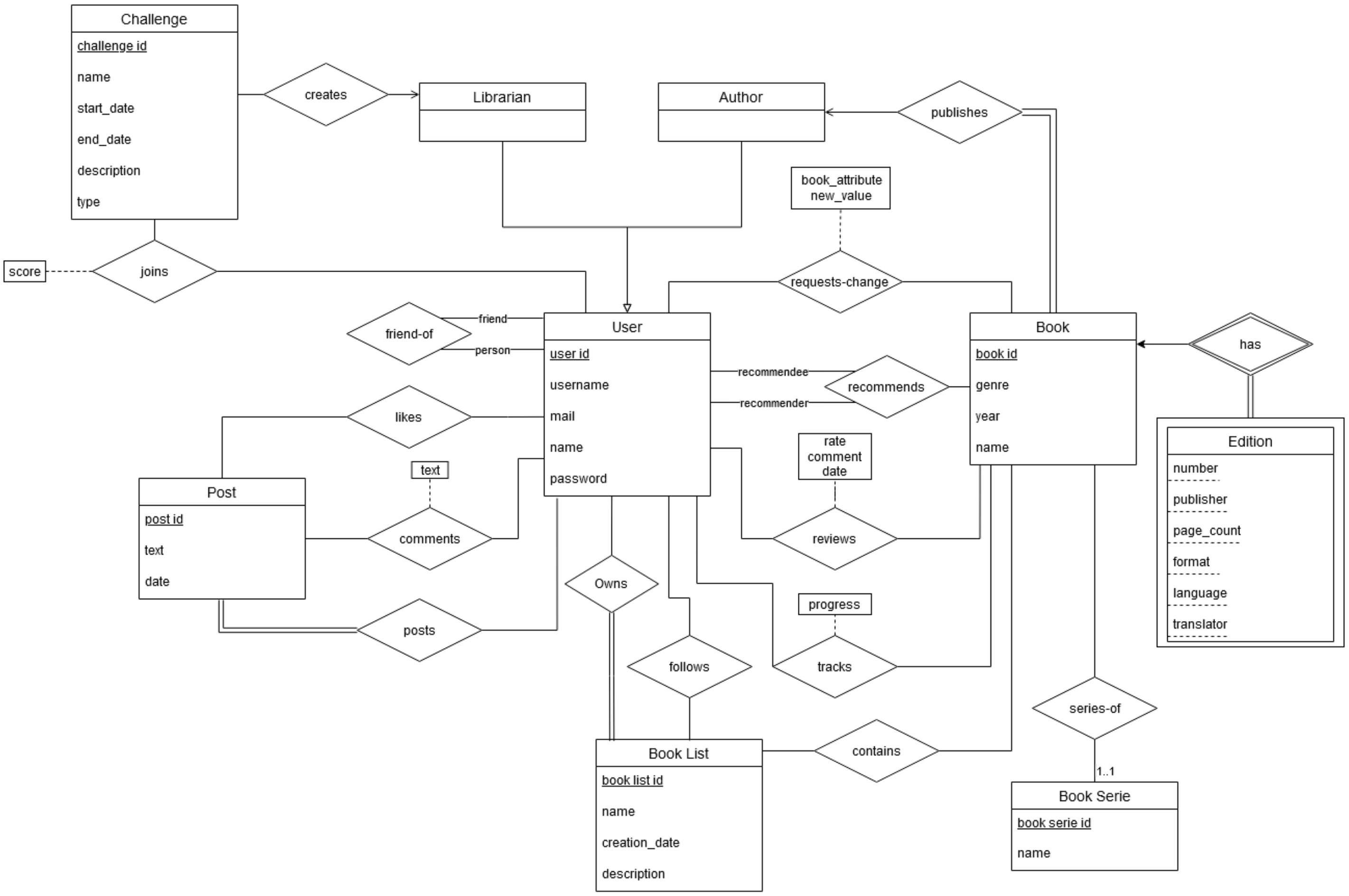
Browser compatibility is an important aspect so that the system can be opened and users can interact with the website on any browser with any version.

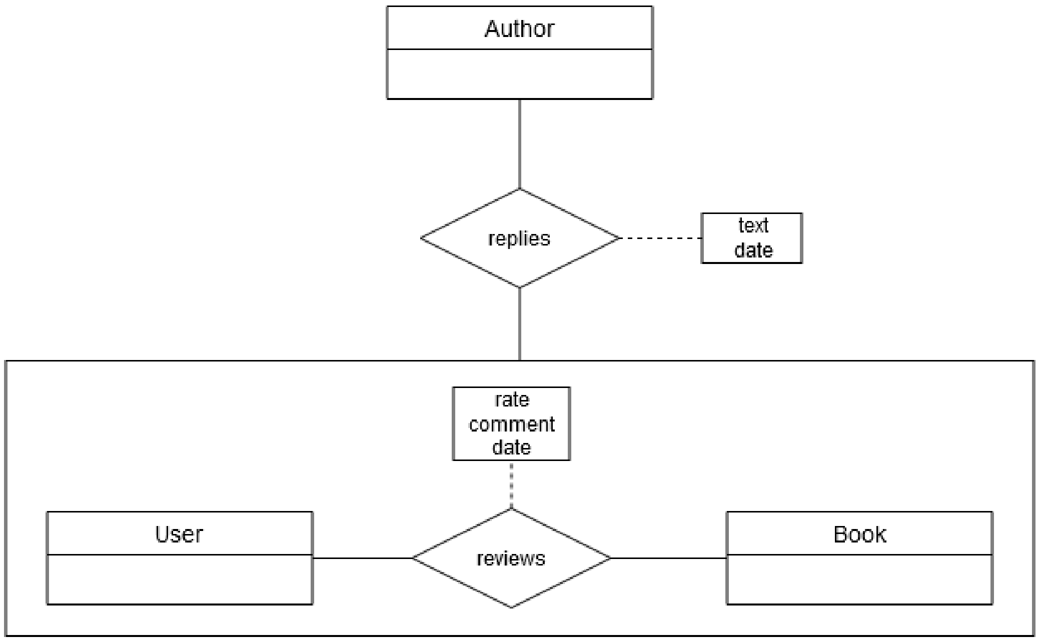
4 Limitations

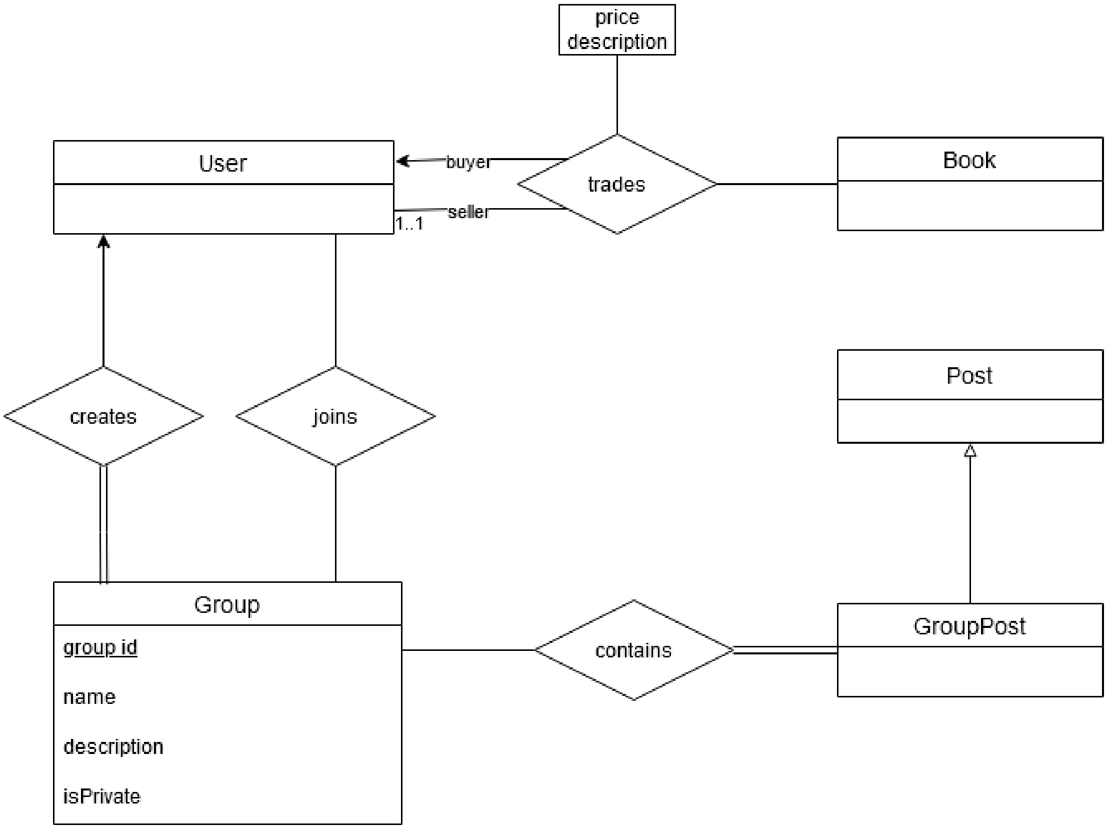
- A book can only be written by a single Author.
- There cannot be a book without any author.
- A user can't mark progress without tracking a book.
- Users can't leave comments without rating the books.
- Rate of user’s review about a book cannot be null.
- There cannot be a book series without any book.
- A book can belong to only one book series.
- Name of the book series cannot be null.
- Name of the book cannot be null.
- Authors cannot publish a book with an existing book_id.
- Users cannot request a books id to be changed.
- Users cannot request to change an attribute of a book to a different domain (varchar to int)
- Name of the book list cannot be null.
- Username and mail of a user are unique attributes.
- None of the attributes of the user can be null.
- Librarians cannot request a change according to specifications (although they are user)
- An account cannot be both librarian and author.
- Authors cannot review their own books.
- Text of the post cannot be null.
- Users cannot comment on posts without any text.
- Name and type of a challenge cannot be null.
- A challenge can be created and organized only by one librarian.

- Authors cannot reply to reviews that don’t belong to their books.
- Authors cannot reply without any text.

5 Conceptual Design, ER Diagram




Author
text
date
replies
rate
comment
date
User
reviews
Book


price
description
User
buyer
seller
1..1
trades
Book
Post
creates
joins
Group
group id
name
description
isPrivate
contains
GroupPost

6 Conclusion

Our project is a web-based application that is a social network platform tailored for book cataloging. In this report, we stated the purpose of this project and provided detailed information about the project. We have stated how to complete this application system completely and clearly together with the technologies we are planning to use. We explained the importance of why and how we should use a database management system and provided a concrete conceptual design. Additionally, we expressed the limitations and requirements of the application. We specified the functional and non-functional requirements for the system. Finally, we created the E / R diagram model for the conceptual definition of the database. We made sure that the database system provides the minimum functionality in the project description and implements additional social features such as being able to sell and buy second hand books from users and create groups to engage in users with similar book taste.

7 Website

CS353 Group Website:
https://zeynepcankara.github.io/Social-Book-Cataloging-Platform/

The Project Repository (private):
https://zeynepCankara/Social-Book-Cataloging-Platform