CNG 352

Database Management System

Games Library Website (Gamebook)



Raed Alsheikh Amin 2528271 || Farnaz Rezaei Noey 2551406

DESCRIPTION:

This web application aims to allow the user to see all games in one click across all the different platforms and stores, such as Steam, EA, Epic Games, etc. In addition, it will allow the user to add friends, add games to preferences, and recommend similar games to the game that the user played before under some conditions. For instance, the user can recommend the game if the game is played 10 hours minimum, and can recommend similar games if he played both games not less than 15 hours each(It may be hard to implement this part). Gamebook application will allow the community of gamers to ask their questions and solve their problems by using Question Response section. The web application will be developed using PHP, and MySQL.

1.1 Data Requirements(assumptions are included)

**User**

GameBook refers to members of the public interested in managing their games across different platforms and stores. To become a User, a person should log in to the website. The data stored on User includes user ID, first name, last name, username, email, password, and date of birth. The user ID is unique for each member. A user can own many Owned Games (Games already bought in each store) and a game can be owned by many users. A user can also have many games as WishList Games and a game can be in many users’ wish list. User also has a recursive relation as Friendship. The data stored on Friendship includes chat and status (with pending or accepted constraints) which belongs to both users.

**Owned Games**

Owned Games are the games that a person has already bought. Owned Games are either installed or not installed. The data stored on Owned Games includes name, icon, description, type, genre, store, age rating, isFavorite, and hours played. Name is the name of the game which is unique across platforms. Age rating has the PEGI3, PEGI7, PEGI12, PEGI16, and PEGI18 constraint. Installed games are stored in the library with their date added.

**Installed**

Installed are the games that the user already has on the PC.

**Not Installed**

Not Installed are the games that the user does not have on the PC but has already bought them.

**WishList Games**

WishList Games refers to the games that the user intends to buy and does not own in any store. The data stored on the WishList Games includes Price, name, icon, description, type, genre, store, and age rating. Name is the name of the game which is unique across platforms. Age rating has the PEGI3, PEGI7, PEGI12, PEGI16, and PEGI18 constraint.

**Library**

The library is where the Owned Games will be stored with their date added. The data stored on Library will include sorting type and Library ID. Library ID is unique for each library. One library can store many Owned Games. Each user has only one library to manage, and that library belongs to only one user.

**Platform**

The data stored on the platform includes platform ID, name, and icon. Platform ID is unique across the platforms. Many Owned Games and many WishList Games belong to one or many platforms.

**Feedback**

The data stored on the Feedback includes Feedback ID, title, comment, timestamp, and rating (which has 0-5 constraint). Feedback ID is unique for each feedback. Feedback is a weak entity because if a user does not make feedback there won’t be any. A user can make many feedback but a feedback belongs to one user, also an Owned Game can have multiple feedback but a feedback belongs to an Owned Game (we cannot give feedback to the games in the Wishlist).

**Question**

Question is where the data related to the questions that the users asked will be stored. The data stored on Question includes QID and text. QID is unique for each question asked. A user can ask many questions, but one question is asked by one user. A question also can have many responses, but a respond belongs to one question.

**Response**

Response is where the data related to the responses to the questions will be stored. The data stored on Response includes ResponseID and text. ResponseID is unique for each response. A user can respond to many questions, but a respond belongs to one user.

1.2 Transaction Requirements

**Data Entry (attributes are simplified but will be implemented fully )**

Enter the details of a new user (such as name: Farnaz, last name: Rezaei, username: farnazrzn, email: [farnazrzn@gmail.com](mailto:farnazrzn@gmail.com), password:1234, DOB: 22/7/2001)

Enter the details of new owned game (such as Minecraft survival genre with PEGI7)

Enter the details of new wish list game (such as Tomb raider adventure genre with PEGI12 for 10$)

Enter the details of new feedback (such as Minecraft bug, sheep got stuck in a wall, with rating of 3)

Enter the details of new question (such as how to make a fishing rod in Minecraft?)

Enter the details of new respond (such as you will need spider webs and wood to make the fishing rod)

Enter the details of new friend (such as Raed Amin with raedamin username and 432 user ID)

**Data update/deletion**

Delete the details of a friend from friends list.

Update/delete the details of a game in the wish list.

Update/delete the details of a game in the owned games list (whether it’s installed or not).

Update/delete the details of the question.

Update/delete the details of the response.

**Data queries**

1. List the details of all friends that the user has.
2. List the details of the friend named Raed.
3. List the details of the pending friends. (status)
4. List the details of all the owned games.
5. List the details of the installed games.
6. List the details of the first-person shooting games. (type)
7. List the details of the favorite games.
8. List the details of the adventure games. (genre)
9. List the details of the games with PEGI12 rating.
10. Group the games based on the belonging platform.
11. List the details of all the games in the wish list.
12. Display the cheapest game on the wish list.
13. Display the game with the highest discount on the wish list.
14. List the details of all the feedback/questions/responses the user made.

2.1 EER Diagram



