

TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Remo Torn 223199IADB

Video Game Backlog Manager

Web Applications with C# project proposal

Supervisor: Andres Käver

Tallinn 2024

Authors declaration of originality

I hereby certify that I am the sole author of this project proposal. All the used materials, references to the literature and the work of others have been referred to. This proposal has not been presented anywhere else.

Author: Remo Torn

21.02.2024

Table of contents

2.	Overview	4
3.	ERD	6
4.	UI.....	7

1. Overview

Video games are a hobby that a lot of people engage with. Thousands of new games are released every year and because of that, a new problem has arisen. There are more games that people want to play than the free time they have on their hands. This project is aiming to deal with that problem by making the choice of what to play easier through data and categorization of chosen games.

Video game backlog is a collection of games, that a player wants to play, but hasn't had the time or energy to do so yet. It usually grows faster than the player can play the games that are already in it. It happens more commonly with single-player games because these are usually an experience the player wants to have, rather than just having fun, like most multiplayer games.

The goal of this project is to create a web application, that would allow people to manage and view information about their video game backlogs. It will provide helpful sorting and searches to help pick a game to play using data about the game entered by the user, like is it owned, how much it costs, how long it is, how excited is the user to play it, has the user interacted it with before and so on.

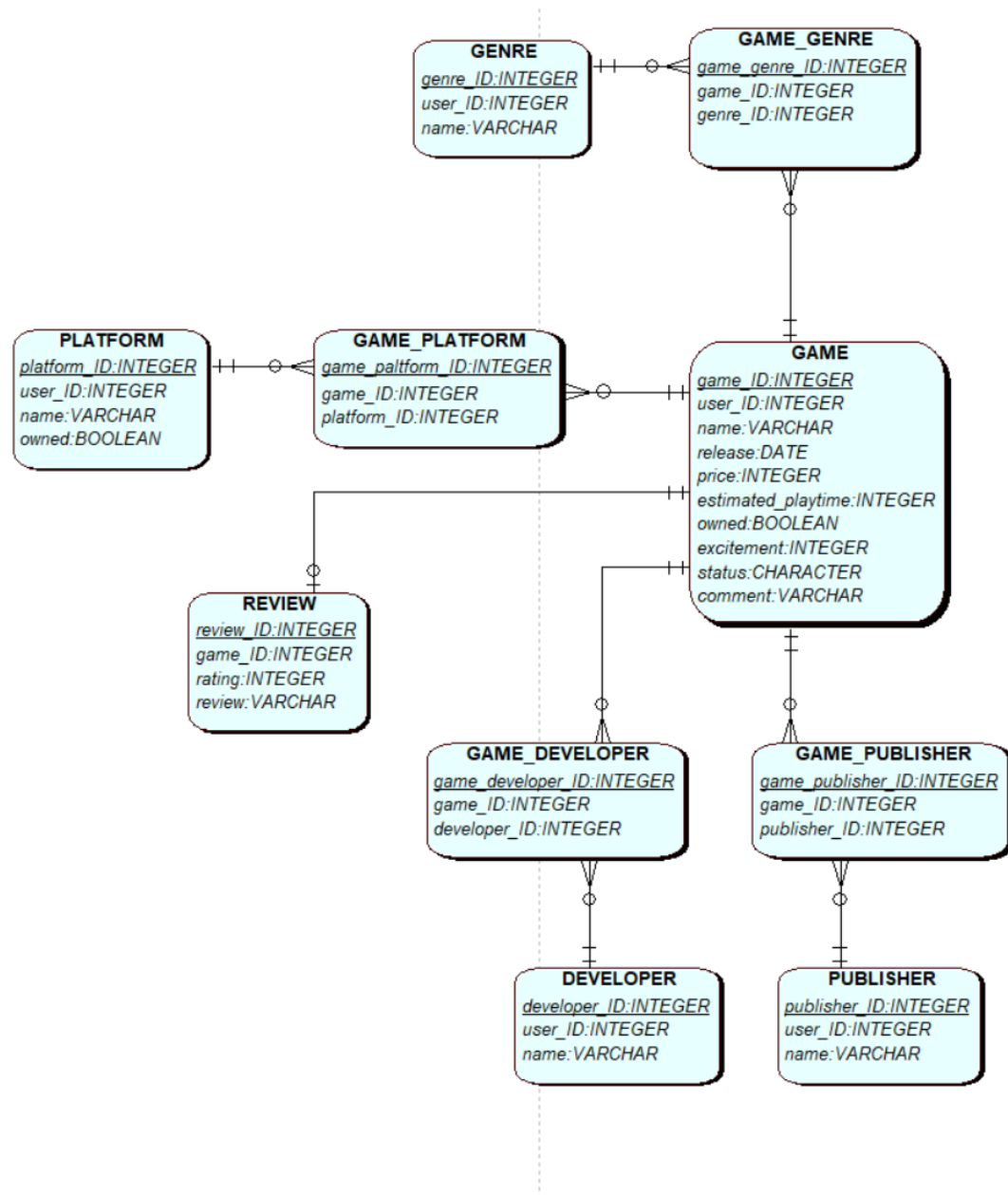
To use the application, user needs to input all the games in his or her backlog to the website. Each game requires the following information about it: game name, its release date to calculate if it is even playable right now, its cost, estimated playtime to know how big of a commitment this game will be, does user already owns it, user excitement to play that game, free form comment about it, its genre or genres, platforms, developers, publishers, and current status in the backlog. Status can be the following: haven't played, don't want to play, playing, finished, dropped, and tried but want to try again.

With all that information, users can sort and filter their games. For example, if a user is busy and doesn't want to spend money, the user can find all games that he or she currently owns, have less than 10 hours of playtime and sort based on excitement. Similar filtering and searching can be done with any data about the game.

The user can also write reviews and give ratings to games. This helps to look back on the finished or dropped games to refresh user's mind about what they liked and didn't like about them. Rating can also be used to get statistics on what type of games the user likes the most.

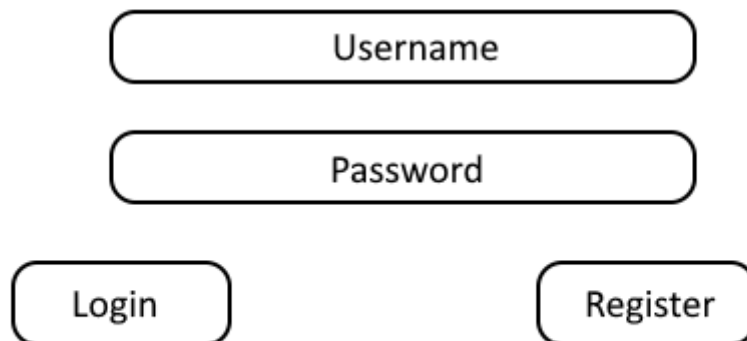
2. ERD

This is the ERD that will be used for the project. It is missing user Identity tables that are provided by the framework. The Foreign Key “user_ID” refers to the Identity “User” Primary Key.



3. UI

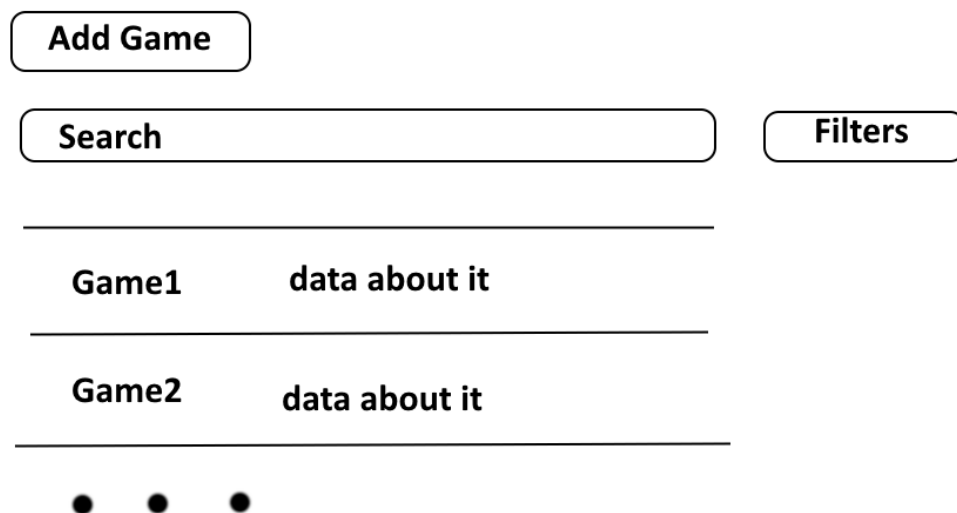
Backlog Manager



A login form with two input fields and two buttons. The first input field is labeled 'Username' and the second is labeled 'Password'. Below the 'Username' field is a 'Login' button, and below the 'Password' field is a 'Register' button.

```
graph TD; Username[Username] --- Password[Password]; Username --- Login[Login]; Password --- Register[Register];
```

Figure 2. Login view



A main view UI diagram. At the top is an 'Add Game' button. Below it is a 'Search' input field and a 'Filters' button. Below these is a table with two rows: 'Game1' and 'Game2', each followed by 'data about it'. At the bottom are three dots.

```
graph TD; AddGame[Add Game] --- Search[Search]; Search --- Filters[Filters]; Search --- Table; Filters --- Table; Table --- Dots[...];
```

Game1	data about it
Game2	data about it

Figure 3. Main view

Game1

Edit

Data1

Data2

Data3



Review

Figure 4. Game view

Game1

Save

Data1

Data2

Data3

...

Review

Figure 5. Add/Edit game view