



Bilkent University

Department of Computer Engineering

CS353 Project Final Report

Video Game Digital Distribution Service

Section 1 / Group 5:

İrem Seven 21704269

Ataberk Gözkaya 21501928

Münevver Uslukılıç 21602297

İlhan Koç 21603429

Instructor:

Özgür Ulusoy

Table of Contents

1.Project Description	4
2.Final E/R	5
3.Relational Schemas	6
3.1 Account	6
3.2 Developer	6
3.3 Publisher	6
3.4 User	6
3.5 Curator	6
3.6 Admin	7
3.7 Game	7
3.8 Mod	7
3.9 Achievements	7
3.10 User-Comment	7
3.11 Curator-Comment	7
3.12 Level	8
3.13 Preferences	8
3.14 ask	8
3.15 develops	8
3.16 updates	8
3.17 adds	9
3.18 publish	9
3.19 follows	9
3.20 user-rates	9
3.21 friendship	10
3.22 wishlist	10
3.23 returns	10
3.24 curator-rates	10
3.25 has-game	10
3.26 has-achievement	11
3.27 make-mod	11
3.28 has-user-mod	11
3.29 has-game-mod	11
4.Advanced Database Features	12
4.1 Views	12
4.2 Stored Procedures	12
4.3 Triggers	12
4.4 Constraints	13
5. Implementation Details	13
6. Sample Output Reports	14

1. Project Description

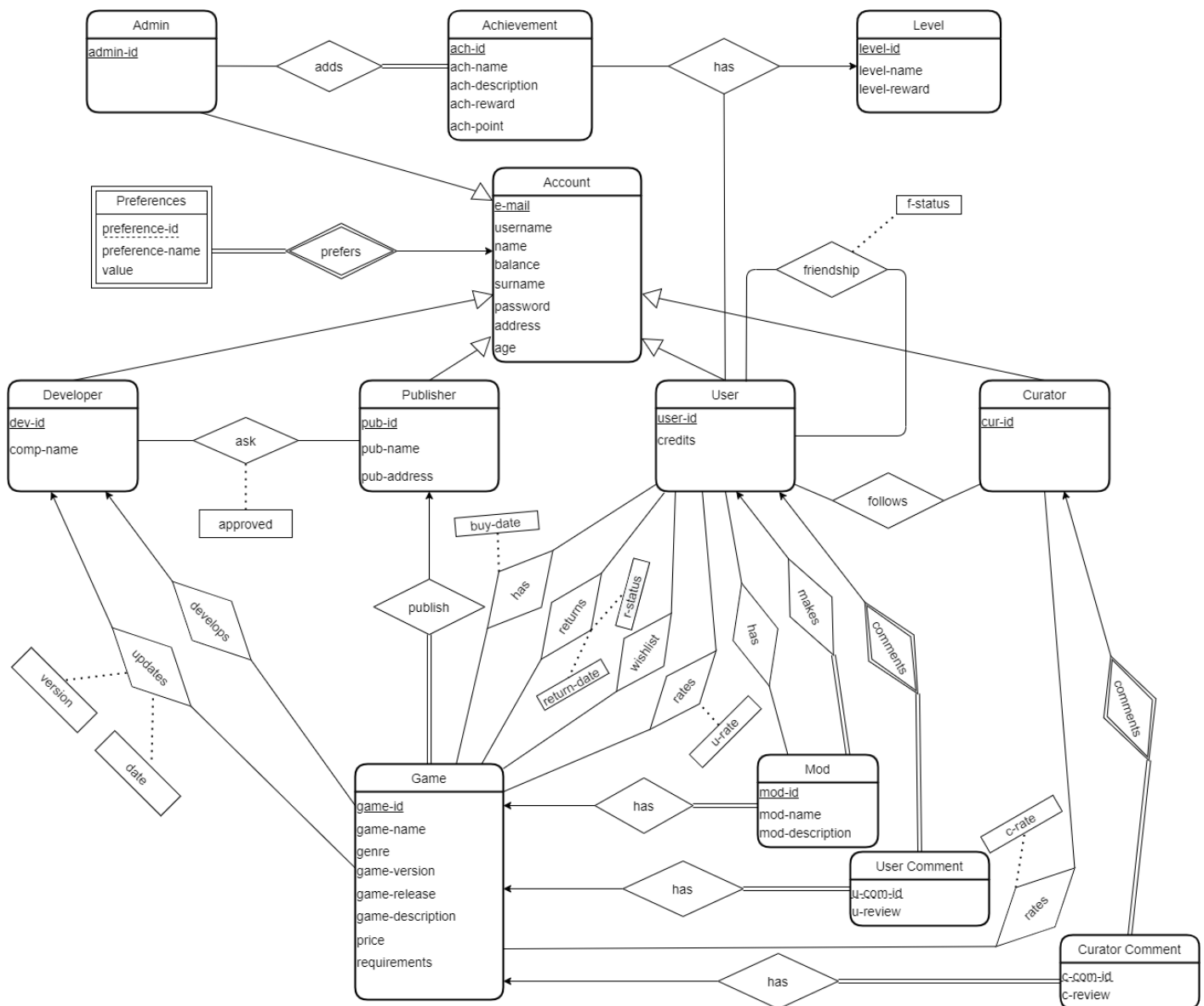
Video Game Digital Distribution System aims to provide a platform that will work as a digital game marketing. It will be a web-based application that aims to provide a safe and useful environment where users can easily get the game they want. Users will be able to browse through the games to discover and download the games they want. If users do not like the game they bought, they have the right to return it. On the other hand, many free games will be available in the system. Users not only will be able to download games but also they will be able to make friends with anyone they want and write comments on the games they want. In the system, there will be curators whose ideas are considered valuable in the gaming industry. Curators will be able to rate games by providing comments on them. As a result of the evaluations provided by the reliable sources here, it will be to ensure that the games that are liked stand out and that the users can access high-quality content. Users will be able to follow the curators they want, thus facilitating their access to the games that suit their game tastes. Some players like to play games as much as they like to develop mods for games, so they can upload mods to the system they have developed for a particular game. They can also download mods for the games that they have bought before.

Not only for users but also we aim to provide a convenient and safe environment for developers and game publishers. Game developers will be able to put their games on the market. However, to maintain the quality in the market, game developers must first obtain permission from the relevant publishing companies to publish the game. After obtaining the necessary permissions from the publishers, the games can be published by the publisher. Publishers may choose to sell games by providing prices on them, or they sell them free. Developers whose games are published will have the authority to update their games.

Entries to the system will be provided with an e-mail address and password. Users can log in directly in this way. On the other hand, developer and publisher companies must provide extra information to register in the system.

Our database provides anything to manage a basic video game distribution service. Users can access various games. Games can be filtered by some specific features in the game store. The database essentially contains all account information of the users, so users can access, modify their account information and look at their games. The database also provides a collection of review data of each game, which users can read and decide whether they want to buy it or not. Thus, as in many other web-applications, using a database system is an important part of our project.

2.Final E/R



3. Relational Schemas

3.1 Account

Relational Model

Account(e-mail, username, name, balance, surname, password, address, age)

3.2 Developer

Relational Model

Developer(dev_id, comp-name, e-mail)

FK: e-mail references Account

3.3 Publisher

Relational Model

Publisher(pub_id, pub-name, pub-address, e-mail)

FK: e-mail references Account

FK: pub-address references Account(address)

3.4 User

Relational Model

User(user_id, credits, e-mail)

FK: e-mail references Account

3.5 Curator

Relational Model

Curator(cur-id, e-mail)

FK: e-mail references Account

3.6 Admin

Relational Model

Admin(admin-id, e-mail)

FK: e-mail references Account

3.7 Game

Relational Model

Game(game-id, game-name, genre, game-version, game-release, game-description, price, requirements)

3.8 Mod

Relational Model

Mod(mod-id, mod-name, mod-description, game-id)

FK: game-id references Game

3.9 Achievements

Relational Model

Achievements(ach-id, ach-name, ach-description, ach-reward, ach-point, level-id)

3.10 User-Comment

Relational Model

User Comment(user-id, u-com-id, u-review, game-id)

FK: user-id references User

FK: game-id references Game

3.11 Curator-Comment

Relational Model

Curator Comment(cur-id, c-com-id, c-review, game-id)

FK: cur-id references Curator

FK: game-id references Game

3.12 Level

Relational Model

Level(level-id, level-name, level-reward)

3.13 Preferences

Relational Model

Preferences(e-mail, preference-id, preference-name, value)

FK: e-mail references Account

3.14 ask

Relational Model

ask(dev-id, pub-id, approved)

FK: dev-id references Developer

FK: pub-id references Publisher

3.15 develops

Relational Model

develops(game-id, dev-id)

FK: game-id references Game

FK: dev-id references Developer

3.16 updates

Relational Model

updates(game-id, dev-id, version, date)

FK: game-id references Game

FK: dev-id references Developer

FK: version references Game(game-version)

3.17 adds

Relational Model

adds(admin-id, ach-id)

FK: admin-id references Admin

FK: ach-id references Achievement

3.18 publish

Relational Model

publish(pub-id, game-id)

FK: pub-id references Publisher

FK: game-id references Game

3.19 follows

Relational Model

follows(user-id, cur-id)

FK: user-id references User

FK: cur-id references Curator

3.20 user-rates

Relational Model

user-rates(game-id, user-id, u-rate)

FK: game-id references Game

FK: user-id references User

3.21 friendship

Relational Model

friendship(user-id, friend-id, f-status)

FK: user-id references User

FK: friend-id references User(user-id)

3.22 wishlist

Relational Model

wishlist(user-id, game-id)

FK: user-id references User

FK: game-id references Game

3.23 returns

Relational Model

returns(user-id, game-id, return-date, r-status)

FK: user-id references User

FK: game-id references Game

3.24 curator-rates

Relational Model

curator-rates(cur-id, game-id, c-rate)

FK: cur-id references Curator

FK: game-id references Game

3.25 has-game

Relational Model

has-game(user-id, game-id, buy-date)

FK: user-id references User
FK: game-id references Game

3.26 has-achievement

Relational Model

has-achievement(user-id, ach-id, level-id)

FK: user-id references User
FK: ach-id references Achievement
FK: level-id references Level

3.27 make-mod

Relational Model

make-mod(mod-id, user-id)

FK: mod-id references Mod
FK: user-id references User

3.28 has-user-mod

Relational Model

has-user-mod(mod-id, user-id)

FK: mod-id references Mod
FK: user-id references User

3.29 has-game-mod

Relational Model

has-game-mod(mod-id, game-id)

FK: mod-id references Mod
FK: user-id references User

4. Advanced Database Features

In this section we show the advanced database components of our project.

4.1 Views

Top 10 cheapest game:

```
CREATE VIEW top_10 AS
SELECT game-name
FROM Game
ORDER BY price DESC
LIMIT 10
```

Games with a particular genre:

```
CREATE VIEW genre AS
SELECT game-name
FROM Game
WHERE genre = @genre
```

Games with a particular publisher:

```
CREATE VIEW publisher-games AS
SELECT game-name,
FROM Game NATURAL JOIN Publisher
WHERE pub-name = @pub-name
```

4.2 Stored Procedures

- Log in and sign up procedures will be stored so we can check for validity during login.

4.3 Triggers

- When a new game is added to the system, the related top 10 cheapest game list can change.
- If a user or a curator rates a game, the rating of the game can be updated.

4.4 Constraints

1. The system cannot be used without an account.
2. The system requires at least a valid email address, username, name, surname, age and a password to be enrolled.
3. An unapproved publisher cannot publish game
4. The password cannot be shorter than 6 characters and cannot be longer than 15 characters
5. Users cannot rate and comment the same game more than one time, but they can edit
6. Curators cannot rate and comment the same game more than one time, but they can edit
7. Each game cannot have more than one publisher
8. A user can add a user only one time
9. A mod can be developed by only one user
10. Users can have only one unique preference which means preferences cannot change device by device.
11. There will be only one payment method which is an application wallet. Users can add money to their wallet and then buy games with money in that wallet.

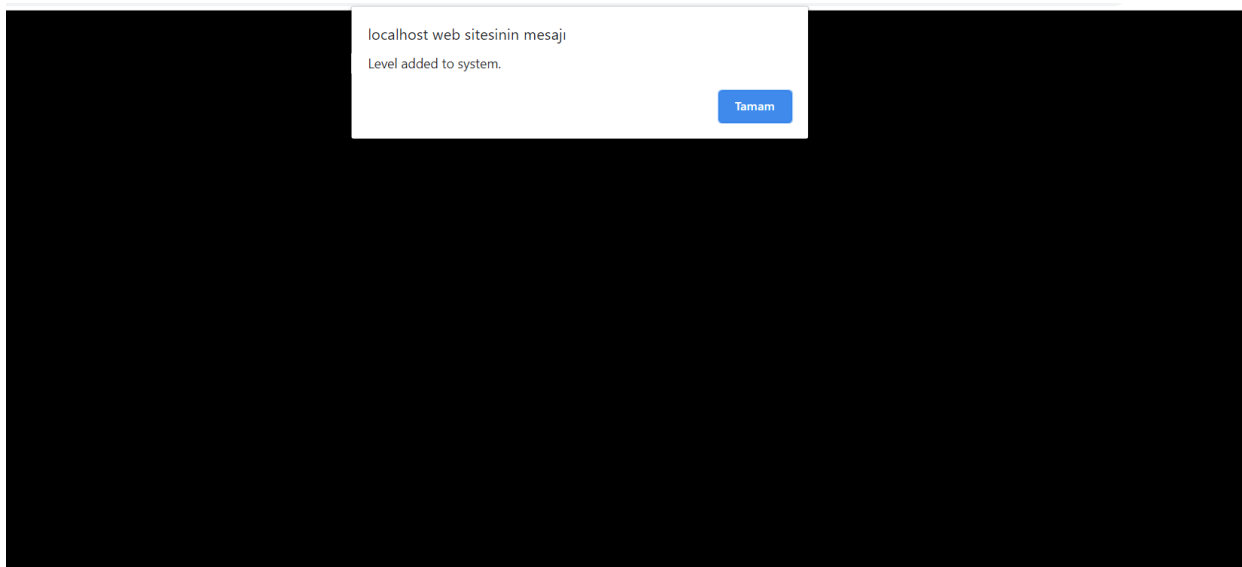
5. Implementation Details

In our project, we used JAVA language to create our database tables by using SQL statements. We connected to our personal mariaDB databases on dijkstra remote machines. After creating the tables on dijkstra database, we manipulated and updated the table data in PHP which has embedded SQL statements. We created a PHP file for each page. For the user interface side we simply used basic HTML with Bootstrap template. By using the Bootstrap template, we changed and designed the pages. We enforced constraints in the database creation phase, which is the initial creation by using our JAVA code. Thus, it enabled us to see errors easily when we update table data. We also used JAVASCRIPT language to be able to control operations more easily and efficiently. For instance, we checked whether the entered fields were empty or not by using user defined functions in JAVASCRIPT.

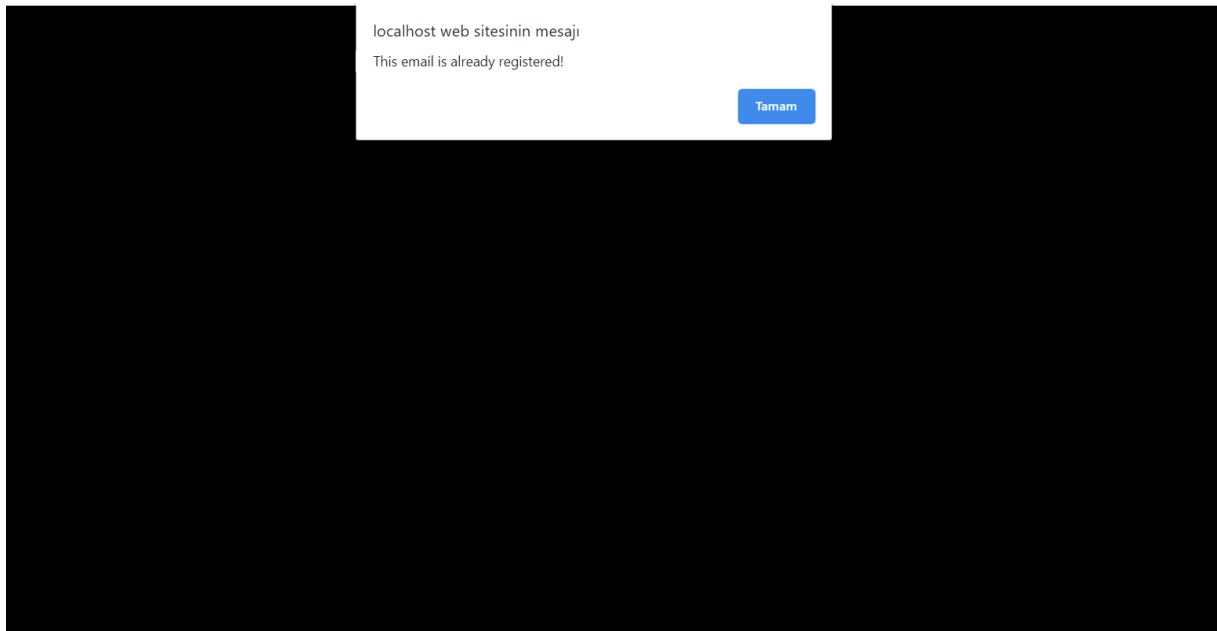
6. Sample Output Reports

The system gives some reports to users for the state of the operations, when the requirements for a specific operation are satisfied or not satisfied by the user inputs. These reports are generally in the form of alert boxes which are shown on the top of the related pages. Some of these examples are given below.

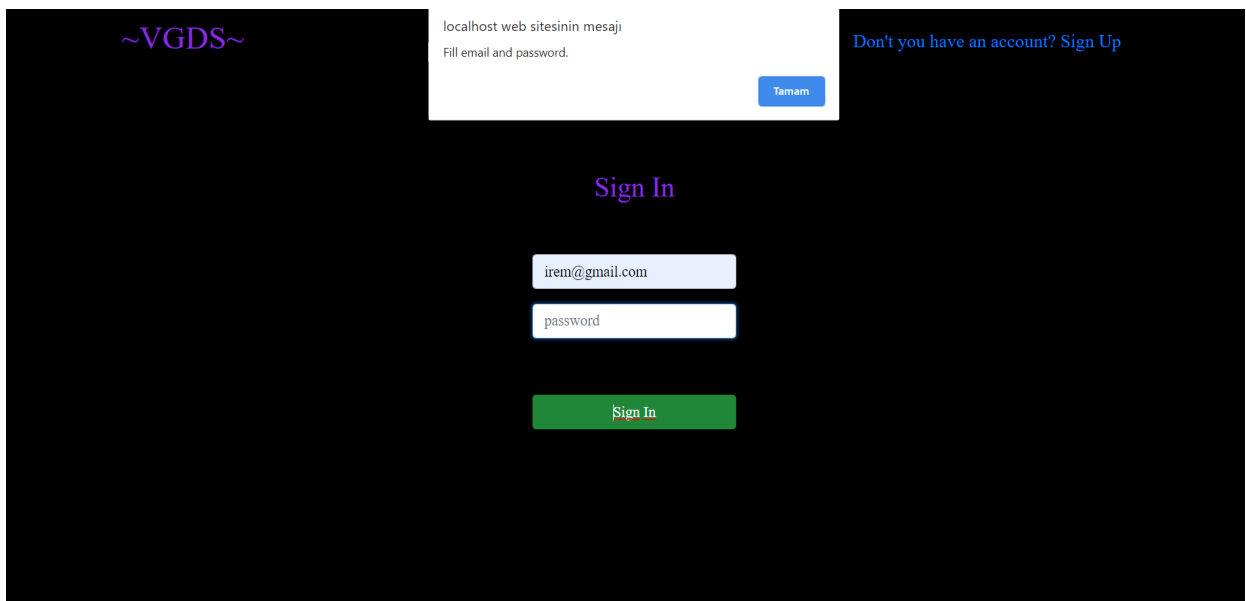
Below it is shown an example alert box report which is given when an admin user adds a new level to the system. Our web-application gives such success reports to the users to inform them the state of their operation.



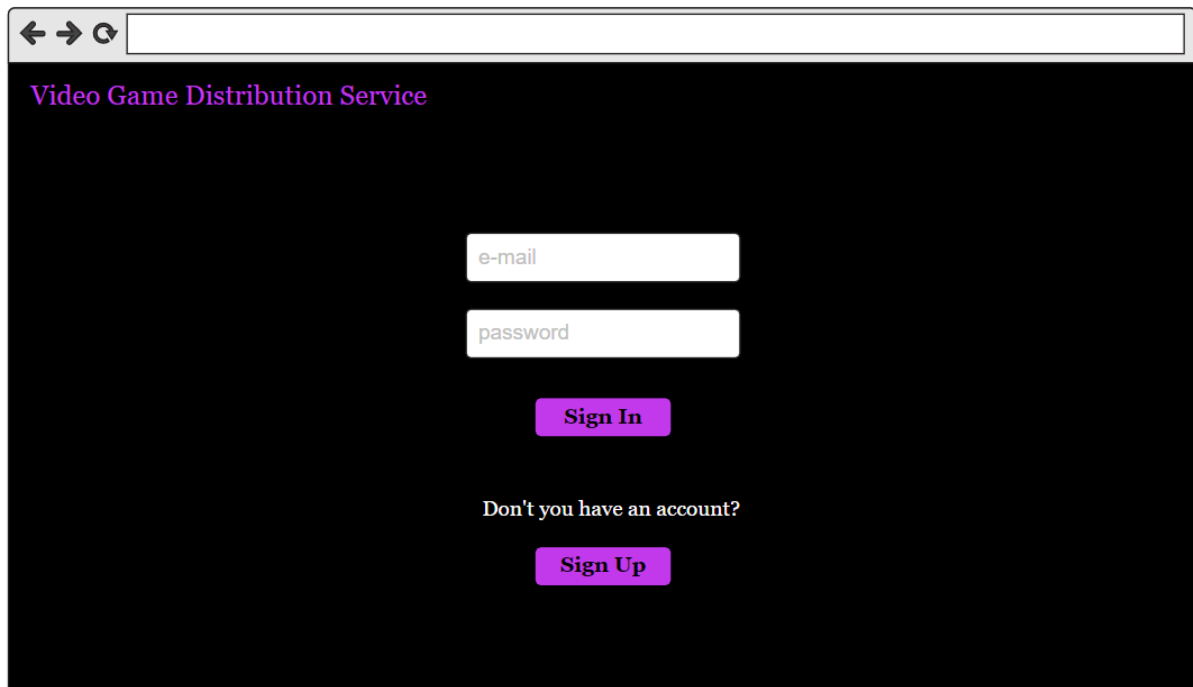
Below it is shown an example alert box report which is shown when the user tries to register the system with an email that is already registered. Our system checks such cases to make sure to protect the uniqueness of the specific data.



Below it is shown an alert box report which is generated when the user does not fill all the required fields to enter the system. Such reports are generated for all the other cases when we need to make sure that we get all the required information by the user inputs.



7. User Manual



The screenshot shows a web browser window with a dark background. At the top, the text "Video Game Distribution Service" is displayed in a light blue font. Below this, there are two white input fields: the first is labeled "e-mail" and the second is labeled "password". Below the input fields is a blue button with the text "Sign In" in white. Below the "Sign In" button is the text "Don't you have an account?" in a light blue font. Below this text is another blue button with the text "Sign Up" in white.

Figure 2 : Sign In / Login Page

Figure 2 is our opening page for the system. It shows the page that lets the user enter the username and the password they have signed up once and when they click the sign in button, the system checks if the username password pair matches and is saved in the system and opens the homepage. If the user does not have the account, he / she can click "Don't you have an account?" button and go to the sign up page.

The image shows a web browser window with a dark blue background. The title bar at the top contains navigation icons (back, forward, refresh) and an empty address bar. The page content is centered and includes the following elements:

- Page Header:** "Video Game Distribution Service" in white text on the left, and a red "Back" button on the right.
- Form Fields:** A vertical stack of seven white input fields with the following labels: "e-mail*", "username*", "name*", "surname*", "address", "age*", and "password*".
- Action Button:** A red "Confirm" button located below the form fields.

Figure 3 : Sign Up Page

Figure 3 shows the sign up page for the users. All of the user types have to enter the email, username, name, surname, age and password necessarily and their address by preferring. If any of the required data is left empty or in a situation that the email is used with another account, the system gives a warning for users. When they click the confirm button, the system opens the Selecting Account Type page after signing up.

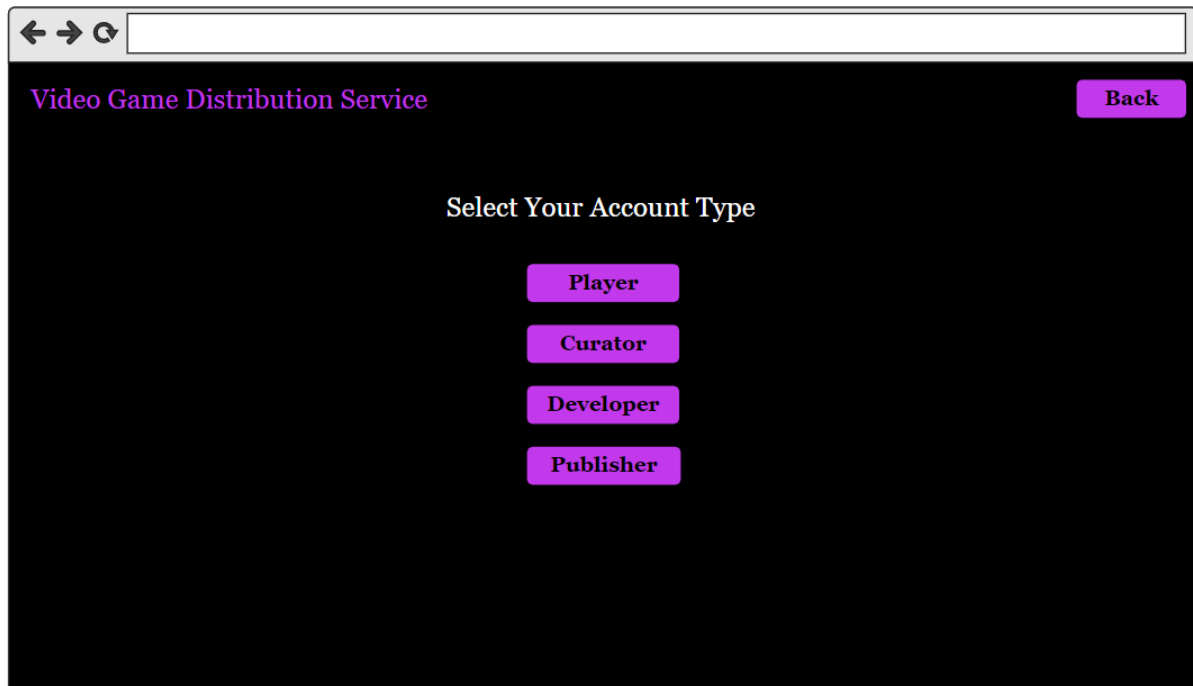


Figure 4: Selecting Account Type

Figure 4 shows the selecting account type page. Since there are different features for all of the account types in the system, all of the users have to choose one of them to use our system. In a situation that they would like to change the information that they have written while signing up, they can click the "Back" button and go back to the Sign Up page.

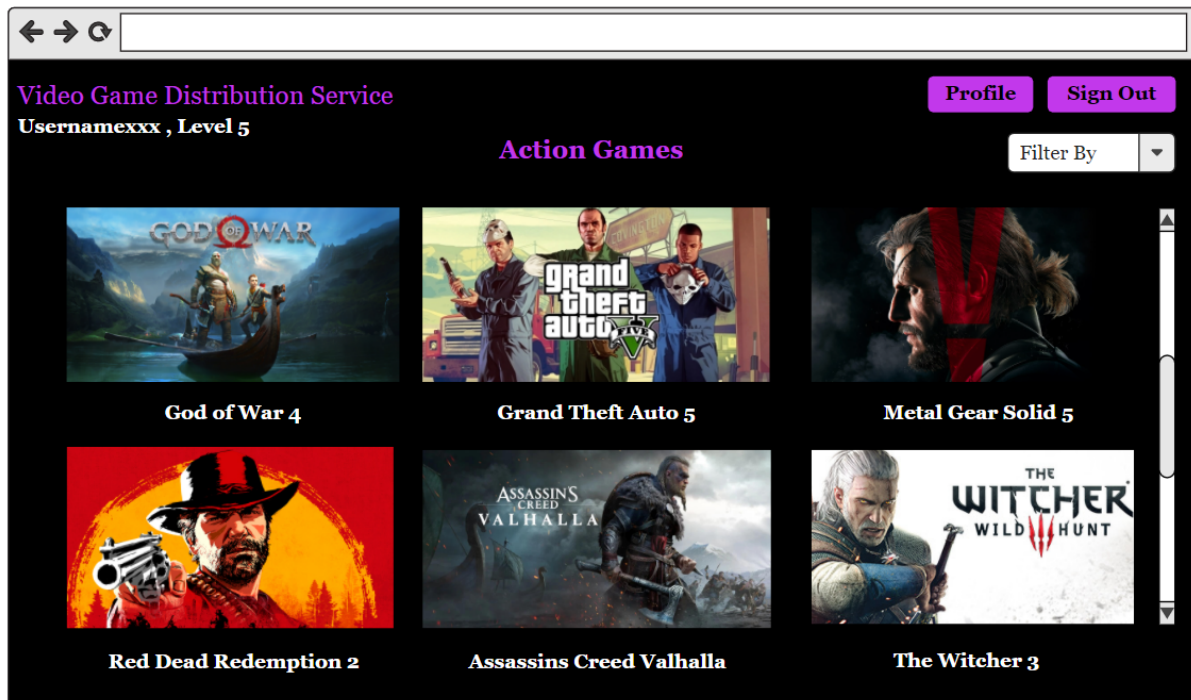


Figure 5: Main Page for All Users

Figure 5 shows the main page for all of the users. When they login with the matching email and password pair, the system opens the game page. At the right corner of the page, there are "Profile" and "Sign Out" buttons. Profile button opens the profile page of the user and Sign Out button let users sign out and leads them to the Login page back. Under those buttons, there is a filter by menu and it shows the genre of the games. Users can click one of the game genres and view the games. They also can click on the games and open the Game View page. In addition to that, beside Player and Curator accounts, all of the users have a special button near the profile button. That button leads developers to the publish request page, publishers to the request vies page, admins to the add achievement page.

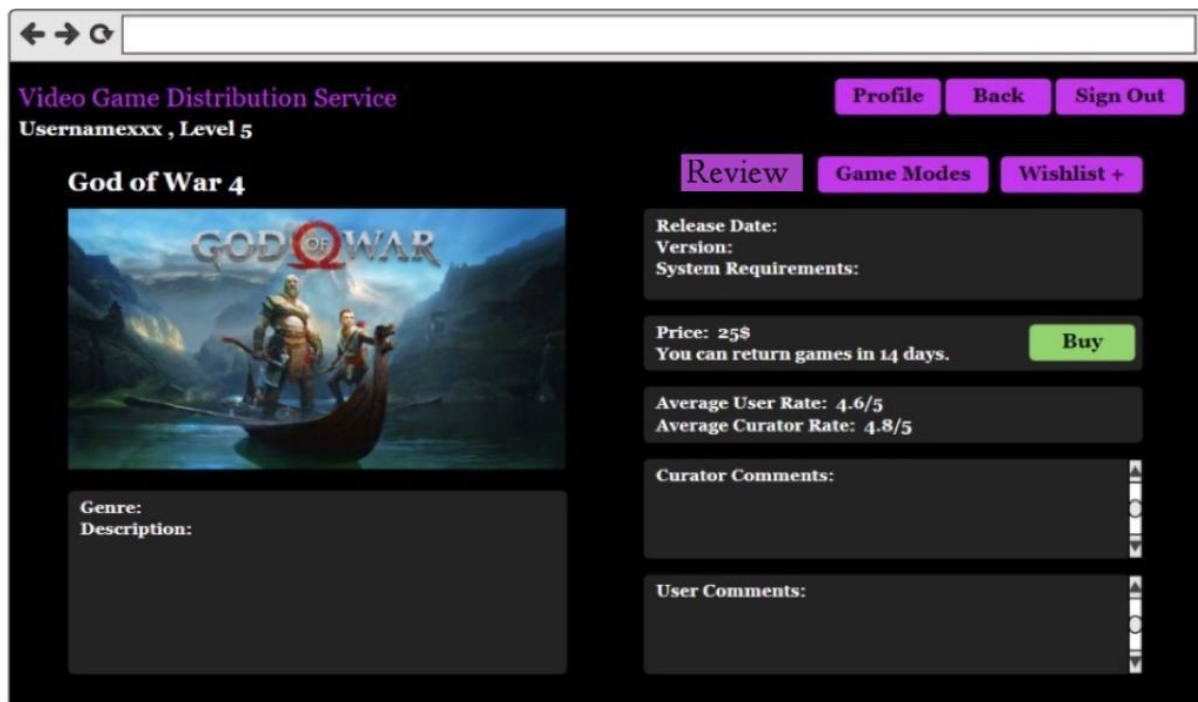


Figure 6 : Game View for Players

Figure 6 shows the Game view for players. On the left corner, there are Review, Game Modes and Wishlist buttons. “Review” button lead players to the page where they can do the rating and the commenting on the selected game. “Game Modes” button shows the game modes of the games, and “Wishlist+” button adds the current game to the wishlist of the user. Other than those, in this page Players can review the genre, game description, release date of the game, version, system requirements, curator comments, user comments, price and the rating of the game. “Buy” button does the payment and downloads the game.

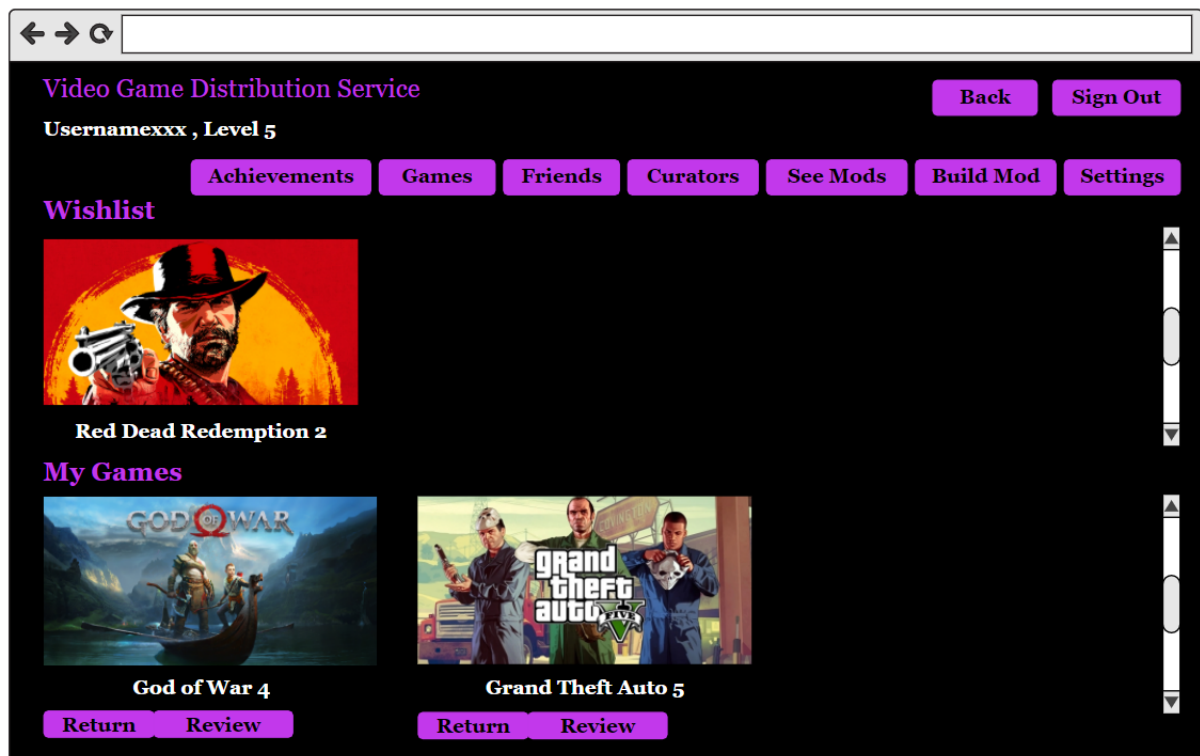


Figure 7: Player Profile

User profile shows the wishlist and purchased games. There are “Return” and “Review” buttons for all of the purchased games. Return button checks whether the returning period has ended or not and does the returning if the returning situation is available. Review button leads Players to the Game Review page. On the top of the page, there are “Achievements”, “Games”, “Friends”, “Curators”, “See Mods”, “Build Mod”, “Settings” buttons for viewing the following pages.

Video Game Distribution Service

UserNamexxx

Built Mode

Select Game

Mod Name

Mod Description

Confirm

Back Sign Out

Figure 8 : Build Mod

Figure 8 shows the Build Mod page for Players. They can choose one of the purchased games with the dropdown menu and enter the mod name and the description and click the “Confirm” button to save it. “Back” button leads the user to the Profile.

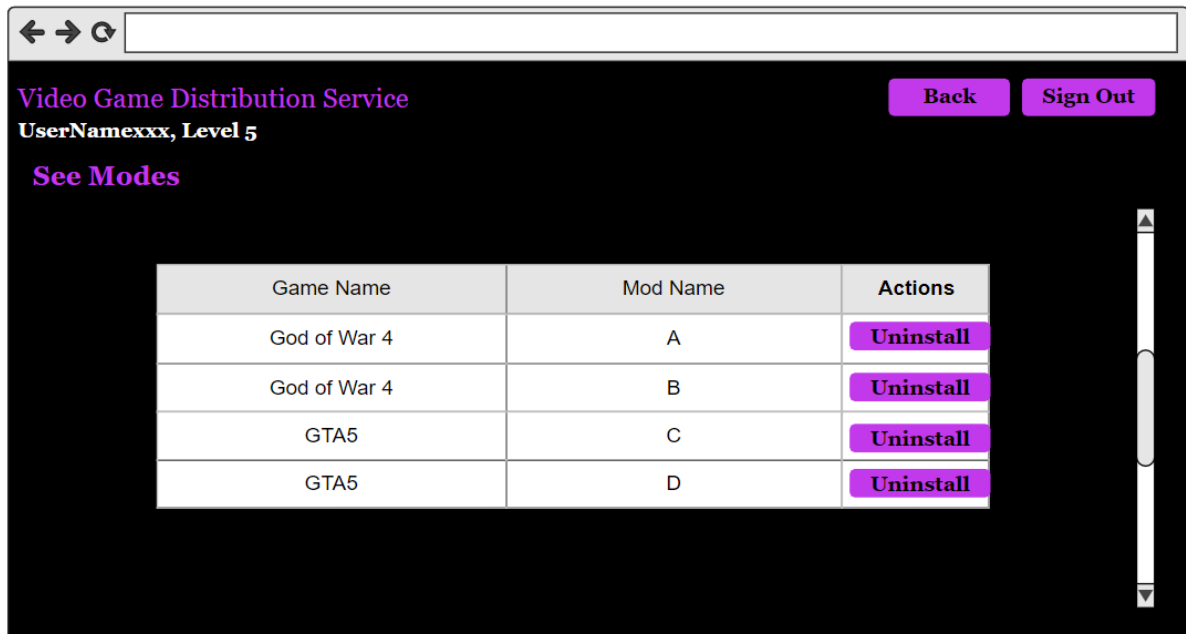


Figure 9: See Modes

Figure 9 shows the installed modes with the Games they are attached and the names of the modes. With the “Uninstall” button, Players can uninstall the mode. The achievements, added friends, following curators are shown in the same layout just as curators.

The screenshot shows a web browser window with a dark background. At the top left, there is a navigation bar with the text "Video Game Distribution Service" in red. To the right of this text are two red buttons: "Back" and "Sign Out". Below the navigation bar, the word "Review" is written in red. On the left side, there is a game cover for "God of War 4" showing Kratos and Atreus in a boat. Below the cover, the text "God of War 4" is written in white. To the right of the cover, there is a form for submitting a review. It includes a "Rate" dropdown menu with a downward arrow, a red "Rate" button, a large white text area for a "Comment", and a red "Comment" button.

Figure 10: Player Game Review

Figure 10 shows the game review page for Players. They can choose the rate from the dropdown menu and click "Rate" to save it. Other than that, they can write comments and click "Comment" to send their comment.

Video Game Distribution Service
Curatornamexxx, Curator

Back Sign Out

Followers: 102

Review

God of War 4

Rate

Rate

Comment

Comment

Figure 11: Game Review for Curators

Figure 11 shows the game review page for curators. They can choose the rate from the dropdown menu and click “Rate” to save it. Other than that, they can write comments and click “Comment” to send their comment. At the left corner of the page, they can view their follower number.

The screenshot shows a web browser window with a dark blue header. On the left, the text "Video Game Distribution Service" is in white, and "DeveloperName" is in yellow. On the right, there are three yellow buttons: "Back", "Sign Out", and "Settings". The main content area has a yellow heading "Request Publish". Below it, there are six white input fields stacked vertically, labeled "Game Name", "Description", "Price", "Genre", "Requirements", and "Publisher Company". At the bottom of the form is a yellow button labeled "Request Publish".

Figure 12: Request Publish for Developers

Figure 12 shows the publication request for developers. They can fill the blanks or game name, description, price, genre, requirements and company and with "Request Publish" buttons the system sends requests to the selected company.

The screenshot shows a web browser window with a dark blue header. On the left, the text "Video Game Distribution Service" is in white, and "DeveloperName" is in yellow. On the right, there are three yellow buttons: "Back", "Sign Out", and "Settings". The main content area has a yellow heading "Update Game". Below it, there are three white input fields stacked vertically. The first is a dropdown menu labeled "Select Game". The second is labeled "Version". The third is labeled "Date". At the bottom of the form is a yellow button labeled "Update".

Figure 13: Upload Game for Developers

Figure 13 shows the updating game for developers. They can upload the game from their computer and add the version and date.

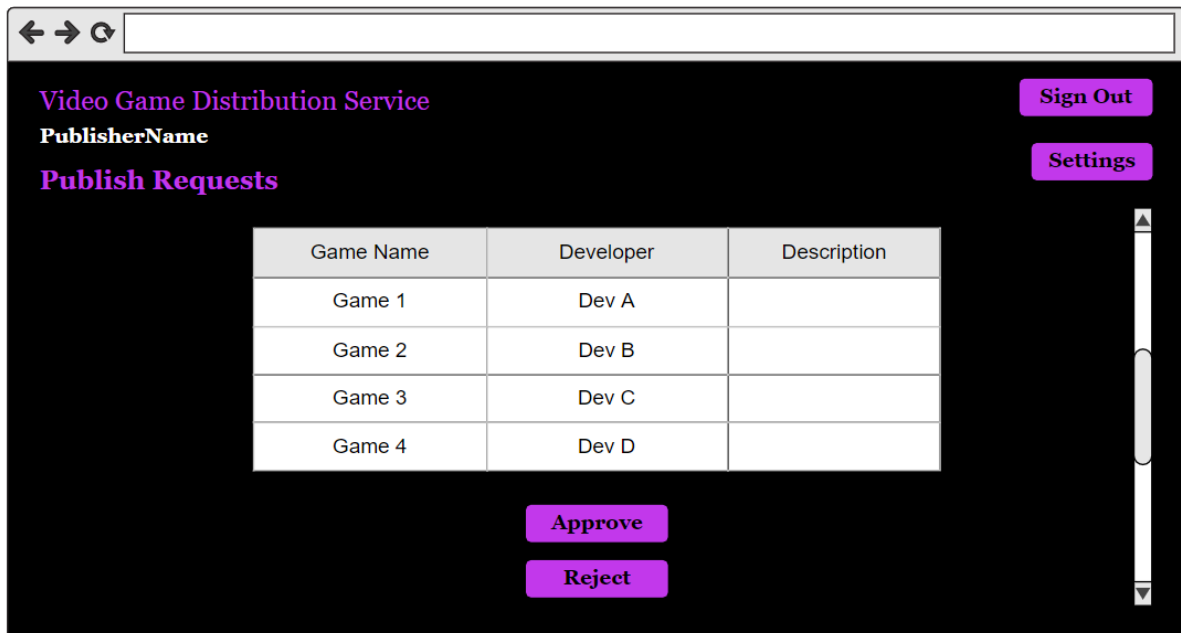


Figure 14: Request View for Publisher

Figure 14 shows the reviewing game publish request has come to the publisher. Publishers can view the developers and descriptions and click the games and choose “Approve” and “Reject” to the request. According to their decision, new games are added to the system and all of the users can view them in the homepage.

~VGDS~

Back Logout

Add Level

level number

level reward

Create

Figure 15: Level Adding for Admin

Admins can add levels to the system, so that they can add achievements to the levels that they have added. They can enter the level number and level reward credits. By clicking the “Create” button, they can create and go to the main admin page.

~VGDS~

Logout

Welcome, admin

Add Level See All Levels

Added Levels

Level Number	Completion Reward	See Achievements	Add Achievement
1	5	See	Add
2	10	See	Add
3	20	See	Add

Figure 16: Seeing Levels for Admin

Admins can view the levels with this page. They can click the see and add buttons and view the achievement description and update the achievement for the related level.

~VGDS~ [Back](#) [Logout](#)

Add New Achievement

Level: 1

name

description

reward points

Add

Figure 17: Add New Achievement for Admin

Admins can add achievements to the levels which they have created previously. To do this they need to click the add button on the right side of the corresponding level. They should provide achievement name, description and the reward points, which will be given to the users when they complete the corresponding level.

~VGDS~ [Back](#) [Logout](#)

Welcome, admin

Achievements

Level: 1

Name	Reward	Description
first game	5	get your first game
first friend	10	get a friend

Figure 18: Achievement List View for Admin

Admins can view the achievements according to their level. They can view the game name, reward and the description of the level that they have chosen from the main admin page.