



Bilkent University

Department of Computer Engineering

CS353-Database Systems

Term Project

Video Game Digital Distribution Service (i.e. Steam)

Final Report

Instructor

Özgür Ulusoy

Assigned TA

Mustafa Can Çavdar

Team Members

Ali Taha Dinçer

İrem Ecem Yelkanat

Muhammed Naci Dalkıran

Sena Korkut

Table Of Contents

Description	4
Contribution of Each Group Member	4
Final E/R Diagram	5
Final Relational Schemas	6
Account	6
User	6
Company	6
Curator	7
Developer Company	7
Publisher Company	7
Wallet	8
Credit Card	8
Video Game	8
Mode	9
Request	9
Subscription Package	9
Subscribes	10
Followed_by	10
Friendship	10
Review	11
Builds	11
Downloads	12
For_m	12
Rates	12
Buys	13
Install	13
Comments_on	13
Include	14
Develops	14
Updates	14
Contains	15
Asks	15
About	15
Publish	16
Takes	16
Implementation Details	17
Advanced Database Components	19

Publisher Company View	19
Reports	20
Users Manual	21
Sign-up Sign-in	22
Regular User Manual	23
Home	23
Wallet Options	23
Store	24
Library	25
Modes	26
Friends	27
Subscription Package Details	28
Video Game Details-Not Bought	28
Video Game Details-Bought	30
Video Game Details-Bought-Installed	31
Build Mode	32
Mode Details	33
Curator Manual	33
Home	33
Video Game Details-Review	34
Developer Company Manual	34
Home	34
Games	34
Create Game	36
Update Game	37
Request	37
Publisher Company Manual	38
Home	38
Requests	38
Approvals	39
Website	40

1. Description

Online game selling platforms are commonly used by people to buy and download games on their computers. With the development of online game selling platforms, gamers have become able to buy newly released games without any overhead for waiting for that game to come to their nearest retailer store and download and play immediately as long as they have an internet connection. Some of them allow users to buy and download games on their computers and some of them allow their users to play any game they want if they pay monthly in a subscription.

Online game selling platforms are important for both gamers, developers and publishers in different manners. With online game selling platforms, gamers started to be able to buy and play their games anytime on any computer, individual developers started to have platforms to sell their games and publishers started to sell games faster without any concern.

The purpose of this project was to implement video game digital distribution service, i.e. Steam and this report considers the final status of the project after implementation.

2. Contribution of Each Group Member

In the process of proposal, design and implementation stages of the project, all group members have worked together and contributed by proposing and improving each other's ideas, implementing functionalities of the platform and debugging each other's parts.

Ali Taha Dinçer: Worked on the implementation of the publisher and developer company user types home pages. Implemented specific functionalities for the regarding user types such as approving, declining a game, creating and updating the game. Implemented admin page and it's functionalities such as showing the result of the “reports” and creating subscription packages.

İrem Ecem Yelkanat: Worked on the implementation of the sign-in and sign-up pages for each user type and the functionality in these pages. Implemented the pages and respective functionalities of the extra feature (subscription packages). Implemented home pages for each user type, transfer money and add credit card pages and their respective functionalities.

Muhammed Naci Dalkıran: Worked on video game functionalities such as buying, returning, installing, uninstalling video games and commenting, reviewing on video games for both regular user and curator types. Moreover, worked on report queries in the admin page.

Sena Korkut: Worked on showing games and modes in the library and modes pages and user functionalities such as building, installing, uninstalling modes for both regular user and curator types. Moreover, implemented Friends page functionalities for both user types.

3. Final E/R Diagram

After the design and implementation process, the previous E/R diagram of the project has altered. Following changes are applied to the system:

- The relationship between Video_Game entity and develops relation is changed into total participation.
- The relationship between Video_Game entity and about relation is changed into total participation.

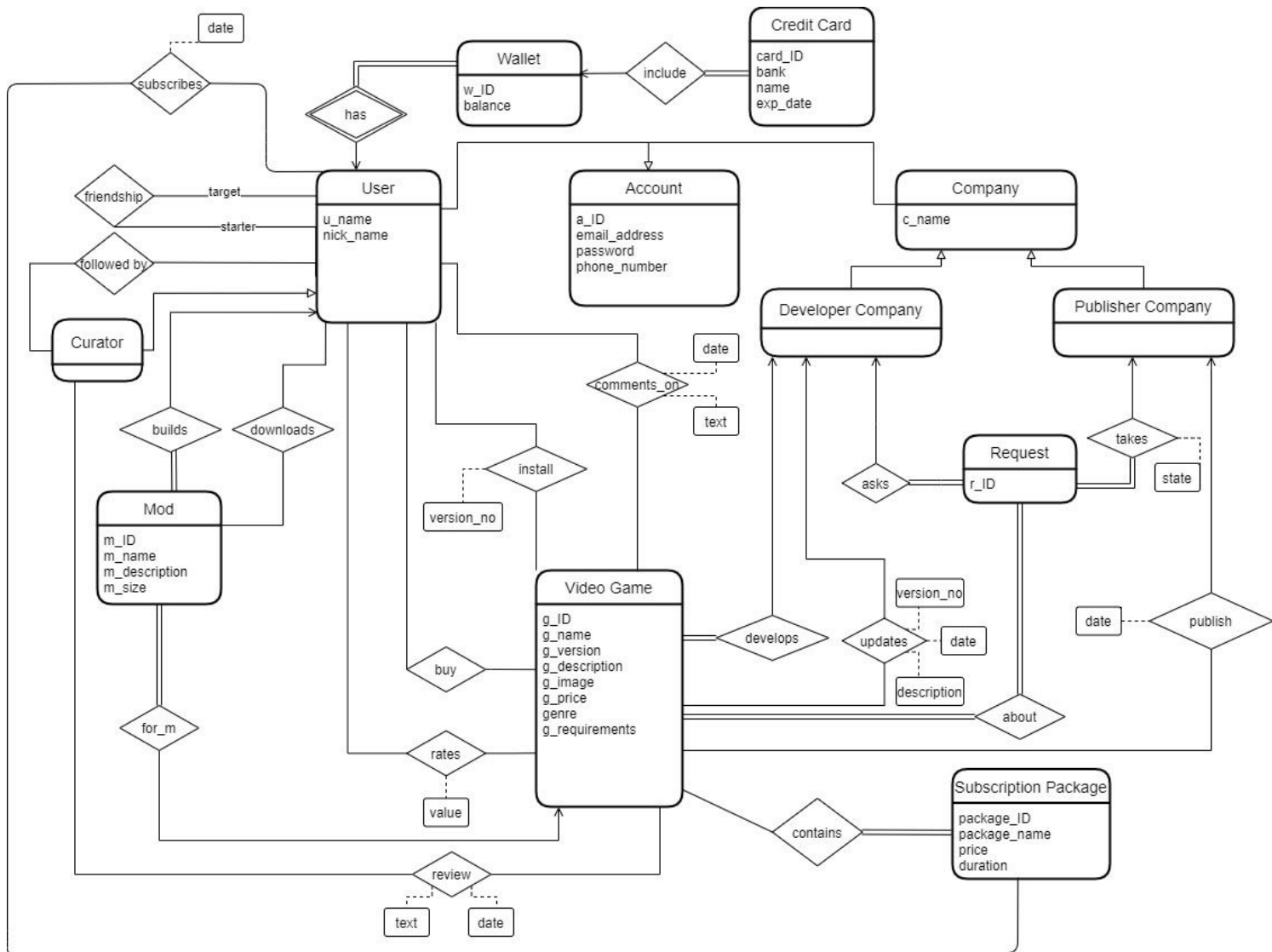


Figure 1: Final E/R Diagram of Video Game Digital Distribution Platform

4. Final Relational Schemas

4.1. Account

Relational Model

Account(a_ID, email_adress, password, phone_number)

Primary Key(s)

a_ID

Foreign Key(s)

-

4.2. User

Relational Model

User(a_ID, u_name, nick_name)

Primary Key(s)

a_ID

Foreign Key(s)

a_ID references Account(a_ID)

4.3. Company

Relational Model

Company(a_ID, c_name)

Primary Key(s)

a_ID

Foreign Key(s)

a_ID references Account(a_ID)

4.4. Curator

Relational Model

Curator(a_ID)

Primary Key(s)

a_ID

Foreign Key(s)

a_ID references User(a_ID)

4.5. Developer Company

Relational Model

Developer_Company(a_ID)

Primary Key(s)

a_ID

Foreign Key(s)

a_ID references Company(a_ID),

4.6. Publisher Company

Relational Model

Publisher_Company(a_ID)

Primary Key(s)

a_ID

Foreign Key(s)

a_ID references Company(a_ID)

4.7. Wallet

Relational Model

Wallet(a_ID, w_ID, balance)

Primary Key(s)

(a_ID, w_ID)

Foreign Key(s)

a_ID references User(a_ID)

4.8. Credit Card

Relational Model

Credit Card(card_ID, bank, name, exp_date)

Primary Key(s)

(card_ID)

Foreign Key(s)

-

4.9. Video Game

Relational Model

Video Game(g_ID, g_name, g_version, g_description, g_image, g_price, genre, g_requirements)

Primary Key(s)

g_ID

Foreign Key(s)

-

4.10. Mode

Relational Model

Mode(m_ID, m_name, m_description, m_size)

Primary Key(s)

m_ID

Foreign Key(s)

-

4.11. Request

Relational Model

Request(r_ID)

Primary Key(s)

r_ID

Foreign Key(s)

-

4.12. Subscription Package

Relational Model

Subscription_Package(package_ID, package_name, price, duration)

Primary Key(s)

package_ID

Foreign Key(s)

-

4.13. Subscribes

Relational Model

subscribes(a_ID, package_ID, date)

Primary Key(s)

(a_ID, package_ID)

Foreign Key(s)

a_ID references User(a_ID)

package_ID references Subscription_Package(package_ID)

4.14. Followed_by

Relational Model

followed_by(c_ID, a_ID)

Primary Key(s)

(c_ID, a_ID)

Foreign Key(s)

a_ID references User(a_ID)

c_ID references Curator(a_ID)

4.15. Friendship

Relational Model

friendship(starter, target)

Primary Key(s)

(starter, target)

Foreign Key(s)

starter references User(a_ID)

target references User(a_ID)

4.16. Review

Relational Model

review(c_ID, g_ID, text, date)

Primary Key(s)

(c_ID, g_ID)

Foreign Key(s)

c_ID references Curator(a_ID)

g_ID references Video_Game(g_ID)

4.17. Builds

Relational Model

builds(m_ID, a_ID)

Primary Key(s)

m_ID

Foreign Key(s)

m_ID references Mode(m_ID)

a_ID references User(a_ID)

4.18. Downloads

Relational Model

downloads(m_ID, a_ID)

Primary Key(s)

(m_ID, a_ID)

Foreign Key(s)

m_ID references Mode(m_ID)

a_ID references User(a_ID)

4.19. For_m

Relational Model

for_m(m_ID, g_ID)

Primary Key(s)

m_ID

Foreign Key(s)

m_ID references Mode(m_ID)

g_ID references Video_Game(g_ID)

4.20. Rates

Relational Model

rates(a_ID, g_ID, value)

Primary Key(s)

(a_ID, g_ID)

Foreign Key(s)

a_ID references User(a_ID)

g_ID references Video_Game(g_ID)

4.21. Buys

Relational Model

buys(a_ID, g_ID, date)

Primary Key(s)

(a_ID, g_ID)

Foreign Key(s)

a_ID references User(a_ID)

g_ID references Video_Game(g_ID)

4.22. Install

Relational Model

install(a_ID, g_ID, version_no)

Primary Key(s)

(a_ID, g_ID)

Foreign Key(s)

a_ID references User(a_ID)

g_ID references Video_Game(g_ID)

4.23. Comments_on

Relational Model

comments_on(a_ID, g_ID, date, text)

Primary Key(s)

(a_ID, g_ID)

Foreign Key(s)

a_ID references User(a_ID)

g_ID references Video_Game(g_ID)

4.24. Include

Relational Model

include(card_ID, w_ID, a_ID)

Primary Key(s)

(card_ID)

Foreign Key(s)

card_ID references Credit_Card(card_ID)

(a_ID, w_ID) references Wallet(a_ID, w_ID)

4.25. Develops

Relational Model

develops(a_ID, g_ID)

Primary Key(s)

g_ID

Foreign Key(s)

a_ID references Developer_Company(a_ID)

g_ID references Video_Game(g_ID)

4.26. Updates

Relational Model

updates(a_ID, g_ID, date, version_no, description)

Primary Key(s)

g_ID

Foreign Key(s)

a_ID references Developer_Company(a_ID)

g_ID references Video_Game(g_ID)

4.27. Contains

Relational Model

contains(package_ID, g_ID)

Primary Key(s)

(package_ID, g_ID)

Foreign Key(s)

package_ID references Subscription_Package(package_ID)

g_ID references Video_Game(g_ID)

4.28. Asks

Relational Model

asks(r_ID, a_ID)

Primary Key(s)

r_ID

Foreign Key(s)

r_ID references Request(r_ID)

a_ID references Developer_Company(a_ID)

4.29. About

Relational Model

about(r_ID, g_ID)

Primary Key(s)

r_ID

Foreign Key(s)

r_ID references Request(r_ID)

g_ID references Video_Game(g_ID)

4.30. Publish

Relational Model

publish(g_ID, a_ID, date)

Primary Key(s)

g_ID

Foreign Key(s)

a_ID references Publisher_Company(a_ID)

g_ID references Video_Game(g_ID)

4.31. Takes

Relational Model

takes(r_ID, a_ID, state)

Primary Key(s)

r_ID

Foreign Key(s)

r_ID references Request(r_ID)

a_ID references Publisher_Company(a_ID)

5. Implementation Details

Like many of the websites running on the internet browsers today, our website also consists of two main parts which are front-end and back-end. Our implementation is started with the back-end part of the project.

With the help of the Java programming language, we wrote a program that builds the database to the Dijkstra server which is provided by Bilkent University. This program connects to Dijkstra server with credentials which is required to access the MySQL server installed in Dijkstra server. After the connection is established successfully, the program continues to create the tables in the database which we provided in the Section 2 of this report. The program also checks if the tables already exist, so that the program can clean the tables and recreate them. Therefore, the program can basically be considered as a cleaner-builder program. During the implementation of the website, we have used a great amount of dummy data and used this program many times to clean the database. In addition, when we have encountered errors due to wrong use of our queries, this program also helped us to start from a blank page. In addition to that, we have used InnoDB as our engine for each table we have created in the database.

After we successfully created the database of the E/R Diagram that we have provided in the Section 2 of this report, we have started to implement the front-end part of the project synchronously with the backend part. We have chosen PHP programming language for server side operations and HTML with in-line CSS for creating the user interface.

The reason we decided to implement front-end and the backend part of the project synchronously was we found that PHP codes can be inserted inside the HTML directly and they are highly related to each other. In order to implement the queries and show their according result as simple and fast as possible, we first created the user interface of the page with HTML and in-line CSS, then we added the PHP codes with respect to the functionality required in the page.

During the implementation of the front-end part synchronously with the backend part, we decided to check the user inputs before sending the form data in order to execute the regarding queries in the database server. The main reason to make that check in the front-end is to avoid sending wrong inputs to server side controllers of the website and additionally, checking the user input in the backend server is a slow process. Therefore, we created simple

JavaScript codes to check the form data provided by the user inputs and we show alert dialogues regarding the cause of the error (if exists) in the user inputs.

The PHP code we have written inside our HTML is divided into two sections. The first section is the PHP codes that we have embedded inside our HTML body. The purpose of these PHP codes are to execute some queries to fetch related data for the purpose of the page and show the data fetched. In order to achieve this, we have used “echo” statements that contain HTML with in-line CSS. By this, for example, we could be able to show the games that were bought by the players in their libraries. The second section of the PHP codes are the server side codes that are executed before the creation of the page. This PHP code checks if the user has signed in or not, gets the session and header information of the page and executes the functions related to the page where these PHP codes implemented if the HTML body of the page contains any form and regarding submit operation. In all the pages of our website, if the user has not signed in, this PHP code section automatically navigates the user to our sign-up page. In addition to that, the both sections of PHP codes navigate users to other related screens. Therefore, we can say that almost all of our functionality used in our website is implemented by using the PHP programming language. The remaining functionalities are the JavaScript codes that check the user inputs from the form submitted by the HTML as discussed in the paragraph above.

As promised in the design report, we have implemented our functionalities regarding different login and sign-up pages for different user types, home pages for different user types; library, store and friends pages for our player and curator user types; approvals and requests pages for publisher company user type and create game and send request pages for developer company user type. We have implemented mode creation and mode showcase, subscription (which is also our extra functionality that we promised in the design report), game showcase and money transfer pages respectfully for player and curator user types. In addition to these pages, we have implemented an extra page for admin, which the core functionality is to create subscription packages and display the results of the complex “report” queries. All of our files are “.php” files which are supported by all of the internet browser applications.

To provide a modern looking and responsive user interface, we have used Bootstrap as our CSS framework. During writing the design report of this website, we have used AdobeXD to create the designs of our website. During the implementation of the website, we have used these designs created in AdobeXD to implement our website with the best possible

user experience. We have used IntelliJ IDEA and VSCode for writing code during the implementation of our website. We have used XAMPP and PHP Server extension for VSCode in order to run our website in our local environment. Finally, we have used DBeaver database management tool to create, manipulate and delete dummy data during testing our website functionalities.

Last but not least, we have encountered one simple problem, which is also a limitation during the implementation process. The problem is the queries with “MINUS” keywords can not be executed in the backend server as MySQL with InnoDB engine does not support “MINUS” keyword. This took a few hours to solve as we can not understand the main reason behind the problem. However, we quickly fixed the problem by changing our queries with the “NOT IN” keyword after we finally found the main reason behind the problem. Additionally, we have enforced our constraints both during the creation of our database tables and front-end checks we implemented with JavaScript code.

6. Advanced Database Components

6.1. Publisher Company View

For showing the games in the Store page, a view is created to avoid displaying games that are created by developer companies but have not been published yet.

```
CREATE VIEW Published_Games AS  
  
SELECT vg.g_ID, vg.g_name, vg.g_version, vg.g_description, vg.g_image, vg.g_price,  
vg.genre, g_requirements  
  
FROM Video_Game vg, publish p  
  
WHERE vg.g_ID = p.g_ID
```

6.2. Reports

Report 1: First report shows the top 5 most bought games within the last 15 days.

```
$newDate = strtotime('-15 days', date("Y-m-d", time( ) ) )
```

```
SELECT *, COUNT(*) AS count_game
```

```
FROM Published_Games vg, buys b
```

```
WHERE b.date > $newDate AND b.g_ID = vg.g_ID
```

```
GROUP BY b.g_ID
```

```
ORDER BY count_game ASC LIMIT 5 ;
```

Report 2: The second report shows the top 5 games that are mostly bought within the last 5 days and have a rate above the average rate of all other games.

```
$newDate = strtotime('-5 days', date("Y-m-d", time( ) ) )
```

```
SELECT *, COUNT(*) AS count_game
```

```
FROM (SELECT *, AVG(r1.value) AS average_rate
```

```
FROM rate r1
```

```
GROUP BY r1.g_ID) AS avg_value_table ,
```

```
SELECT *, AVG(r2.value) AS r2_avg
```

```
FROM rate r2
```

```
GROUP BY r2.g_ID) AS r2_table, Published_Games vg, buys b
```

```
WHERE b.date > $newDate AND b.g_ID = vg.g_ID AND r2_table.g_ID = vg.g_ID
```

```
GROUP BY r2_table.g_ID
```

```
HAVING AVG(avg_value_table.average_rate) < r2_table.r2_avg
```

```
ORDER BY count_game ASC LIMIT 5 ;
```

7. Users Manual

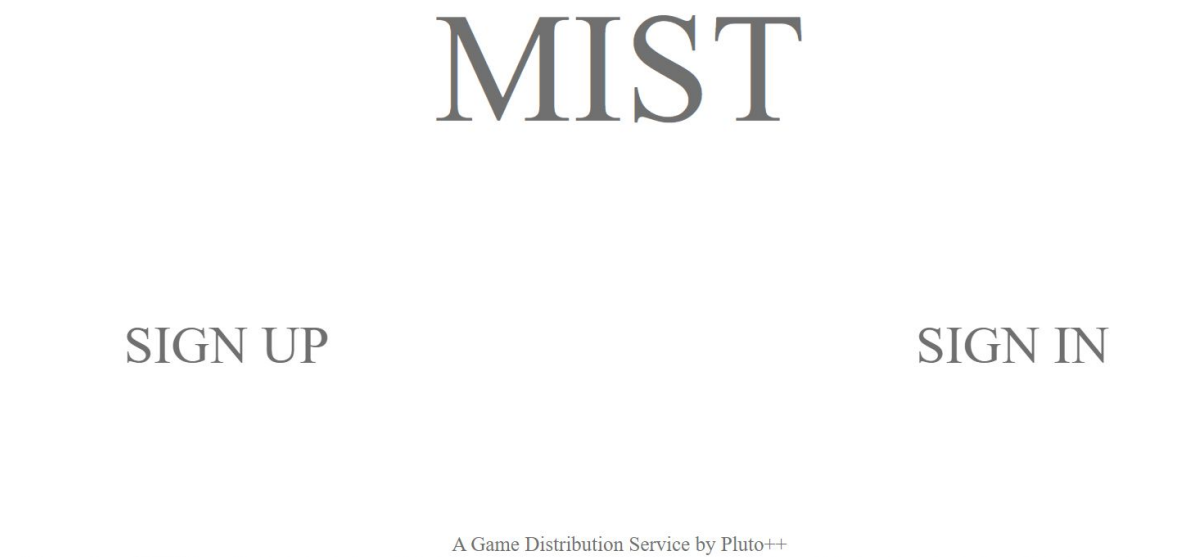


Figure 2: Main Screen of the Website

When the application is opened, the opening screen in *Figure 2* is displayed. User can either choose to sign in or sign up, if one has not registered before.

7.1. Sign-up Sign-in

Sign Up

User

Curator

Dev. Co.

Pub. Co.

@ Full Name

@ Nick Name

@ Email

@ Password

@ Phone Number

Sign Up

A Game Distribution Service by Pluto++

Sign Up

User

Curator

Dev. Co.

Pub. Co.

@ Company Name

@ Email

@ Password

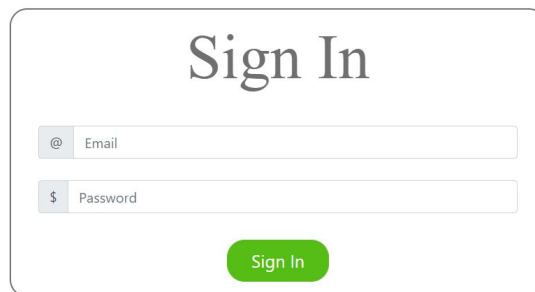
@ Phone Number

Sign Up

A Game Distribution Service by Pluto++

Figure 3: Sign-up Requirements for User and Company Types

In sign up screen, user first chooses the type of the account, i.e. User, Curator, Developer Company and Publisher Company. If the account is a type of User, full name and nickname along with e-mail address, password and phone number are asked. If the account is a type of Company, company name along with e-mail address, password and phone number are asked. Above screenshots show user (regular user and curator) and company (publisher and developer companies) information for signing up.



The image shows a 'Sign In' form. It has a title 'Sign In' at the top. Below it are two input fields: one for 'Email' with an '@' icon and one for 'Password' with a '\$' icon. At the bottom is a green button labeled 'Sign In'.

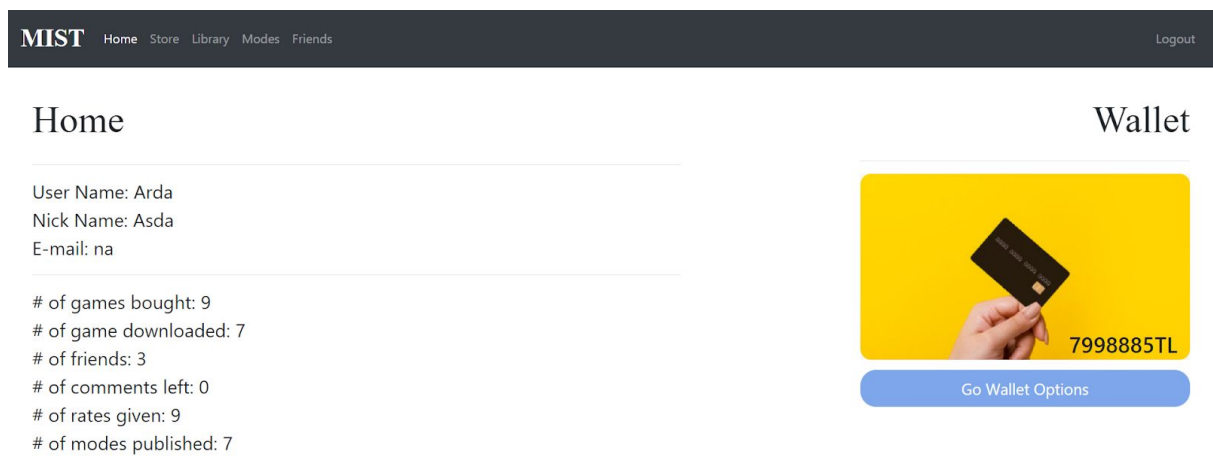
A Game Distribution Service by Pluto++

Figure 4: Sign in Screen

After signing up or choosing sign in from the main screen, one should enter e-mail address and password to sign in to the system.

7.2. Regular User Manual

7.2.1. Home



The image shows the 'Home' page of a user. At the top is a dark navigation bar with 'MIST' and links to 'Home', 'Store', 'Library', 'Modes', and 'Friends'. A 'Logout' link is in the top right. The main content is split into two columns. The left column, titled 'Home', shows user details: 'User Name: Arda', 'Nick Name: Asda', and 'E-mail: na'. Below this is a list of statistics: '# of games bought: 9', '# of game downloaded: 7', '# of friends: 3', '# of comments left: 0', '# of rates given: 9', and '# of modes published: 7'. The right column, titled 'Wallet', features a yellow background with an image of a hand holding a black card. The card has the number '7998885TL' on it. Below the image is a blue button labeled 'Go Wallet Options'.

A Game Distribution Service by Pluto++

Figure 5: Home Page for User

In the home page, shown above *Figure 5*, several different information about the user and the activity is shown along with the wallet and its balance. From the upper left side, the user can change between pages for different procedures.

7.2.2. *Wallet Options*

The screenshot shows the 'Add Credit Card' section of the MIST application. At the top, there is a navigation bar with 'MIST' and links to 'Home', 'Store', 'Library', 'Modes', and 'Friends'. A 'Logout' link is in the top right. Below the navigation bar, there are two buttons: 'Add Credit Card' (highlighted in blue) and 'Transfer Money'. The main content area is titled 'Current Credit Cards' and displays a list of five credit cards with their names and numbers. Below this list, there are input fields for 'Name of the Card', 'Bank Name', and 'Card Number'. A green 'Add New Card' button is at the bottom of the form.

Card Name	Card Number
irem	1111111111111111
keke	1212121212121212
sema	1212121212121213
ali	8888888888888888
e	xxxxxxxxxxxxxxxx

Input fields: Name of the Card, Bank Name, Card Number (with a placeholder 'gg aa yyyy').

Buttons: Add Credit Card, Transfer Money, Add New Card.

Figure 6: Add Credit Card Section

After clicking wallet options, the wallet page is displayed as in *Figure 6*. Users can add credit cards for transferring money to the wallet. Current credit cards and fields for new credit cards are displayed.

The screenshot shows the 'Transfer Money' section of the MIST application. At the top, there is a navigation bar with 'MIST' and links to 'Home', 'Store', 'Library', 'Modes', and 'Friends'. A 'Logout' link is in the top right. Below the navigation bar, there are two buttons: 'Add Credit Card' and 'Transfer Money' (highlighted in blue). The main content area is titled 'Current Credit Cards' and displays a list of five credit cards with their names and numbers. Below this list, there is an input field for 'Amount'. A green 'Transfer' button is at the bottom of the form.

Card Name	Card Number
irem	1111111111111111
keke	1212121212121212
sema	1212121212121213
ali	8888888888888888
e	xxxxxxxxxxxxxxxx

Input field: Amount.

Buttons: Add Credit Card, Transfer Money, Transfer.

Figure 7: Transfer Money Section

After choosing the transfer money tab, the page is displayed in *Figure 7*. User chooses a credit card and enters the amount of money to transfer money to the wallet.

7.2.3. Store

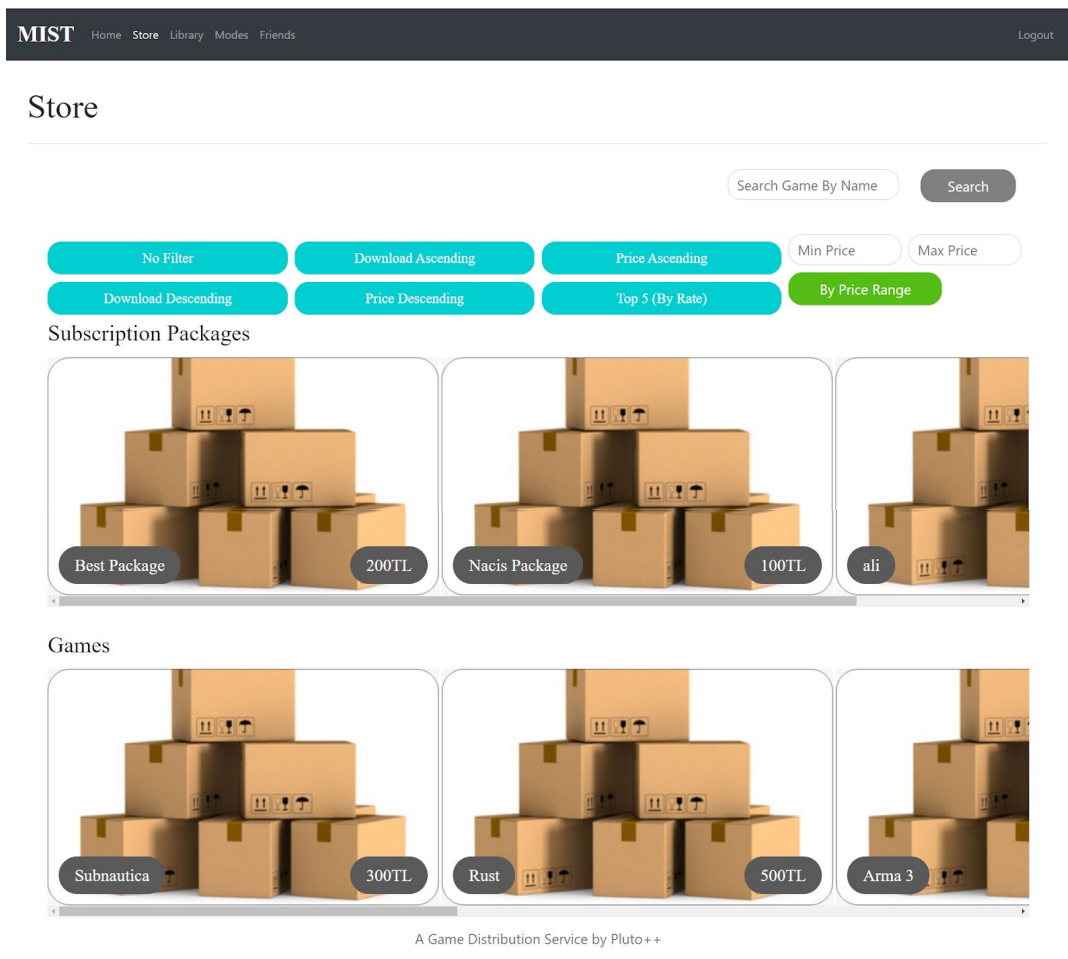


Figure 8: Store Page

Store page shows subscription packages and all games ordered by their rates. In store, the name and the price of packages and games are displayed along with images. At the upper right corner of the store page, there is a search button to search games according to their game names. Below the search bar, there are different options to filter and order games according to their different attributes. Default store page is with no filter, however, one can display the games in ascending or descending order according to price or number of downloads. One can also display top 5 high rated games.

7.2.4. Library

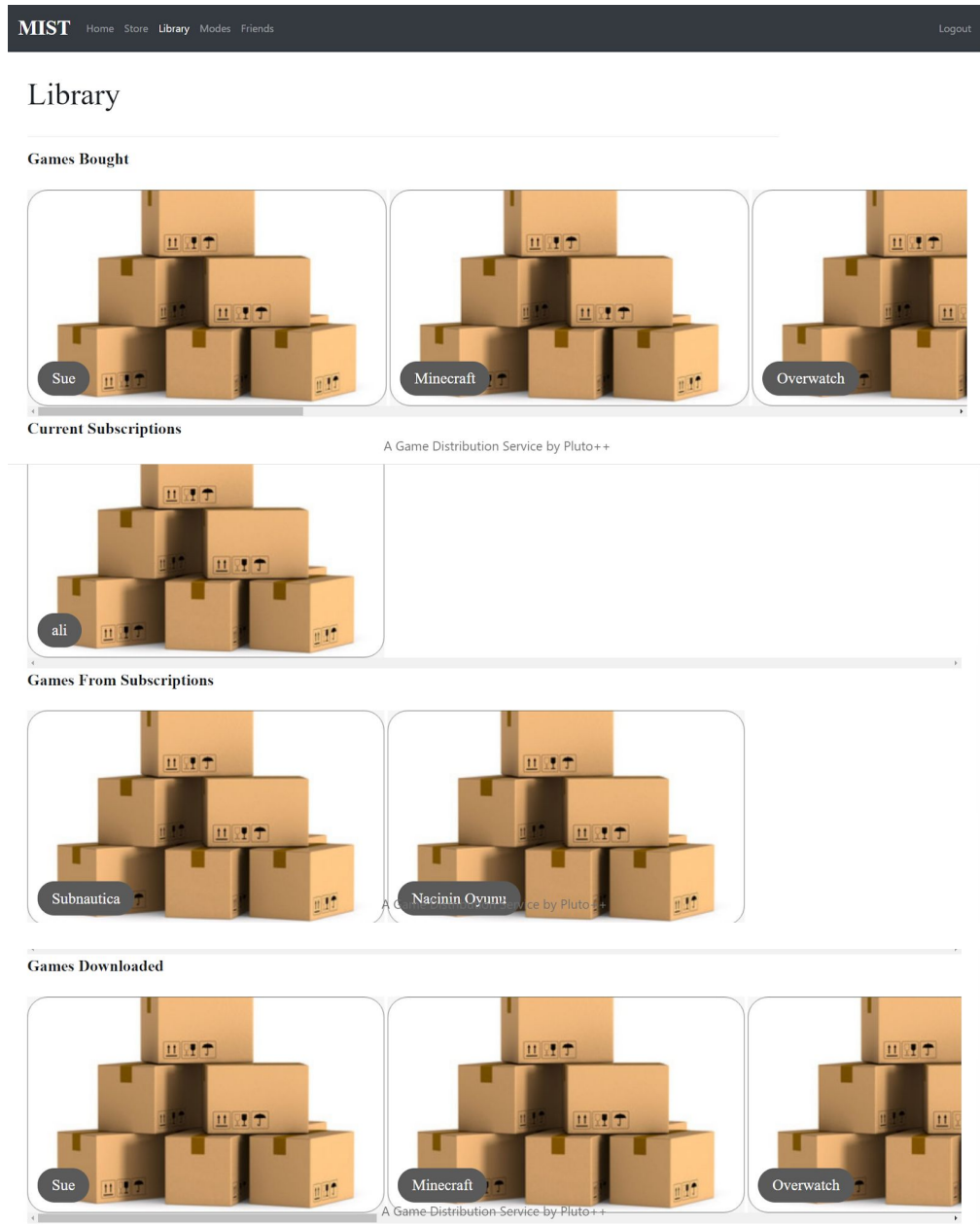


Figure 9: Library Page

Library page shows the games bought, downloaded and came from subscription packages. Additionally, subscribed subscription packages are shown as in *Figure 9*.

7.2.5. Modes

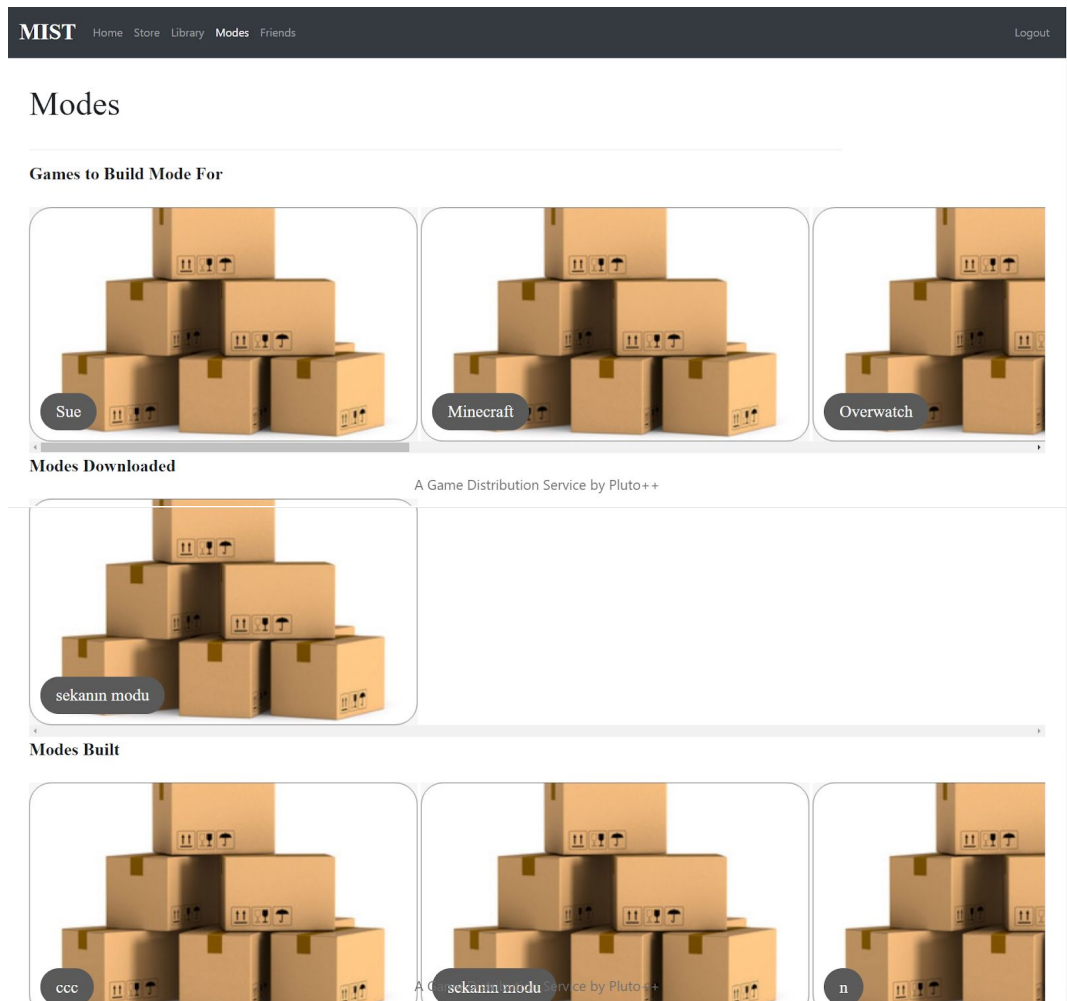


Figure 10: Modes Page

In the Modes page, users can see their games that they can build a mod for, which are the ones the user bought, and the mods they have installed before as in Figure 31.

7.2.6. *Friends*

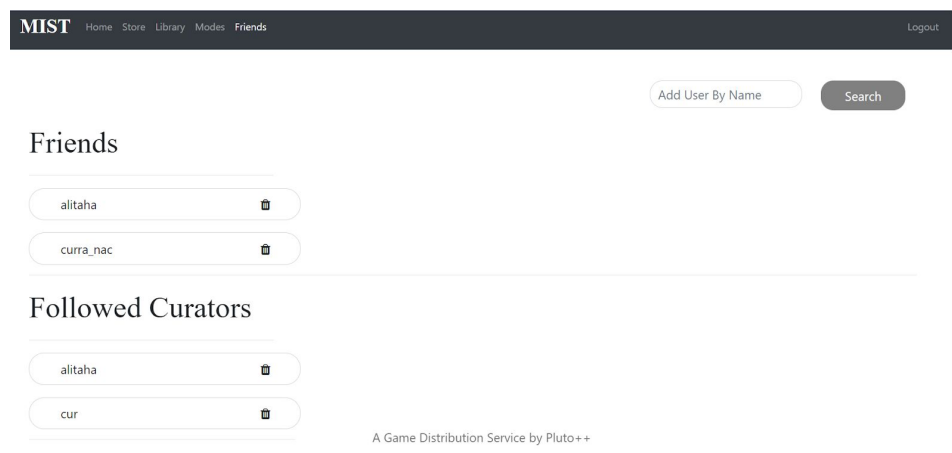


Figure 11: Friends Page

In the Friends page, user types can see their friends and the curators they are currently following as can be seen in *Figure 11*. Users and Curators can unfollow a friend or a curator by pressing the trash can button.

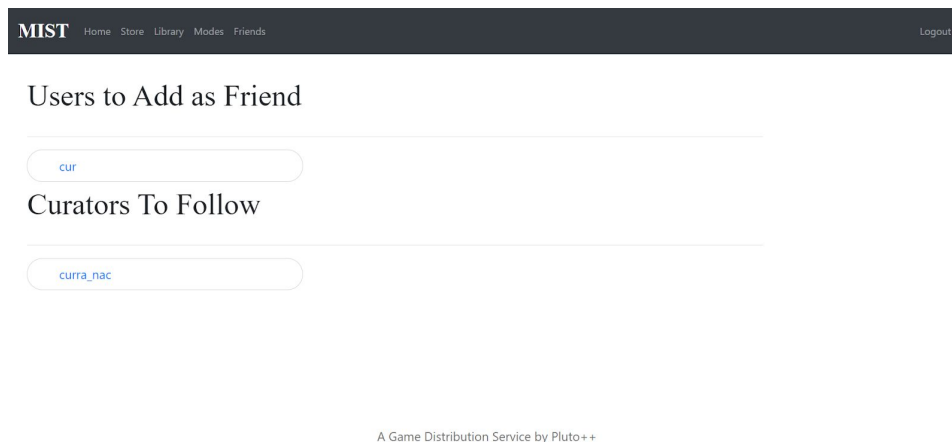


Figure 12: Search Results Page in Friends Page

From the search field, users can search a word to find related users and curators for adding as friends or following as can be seen from the example page in *Figure 12*. Clicking on the nicknames of users is enough to add them as friends or follow them.

7.2.7. Subscription Package Details

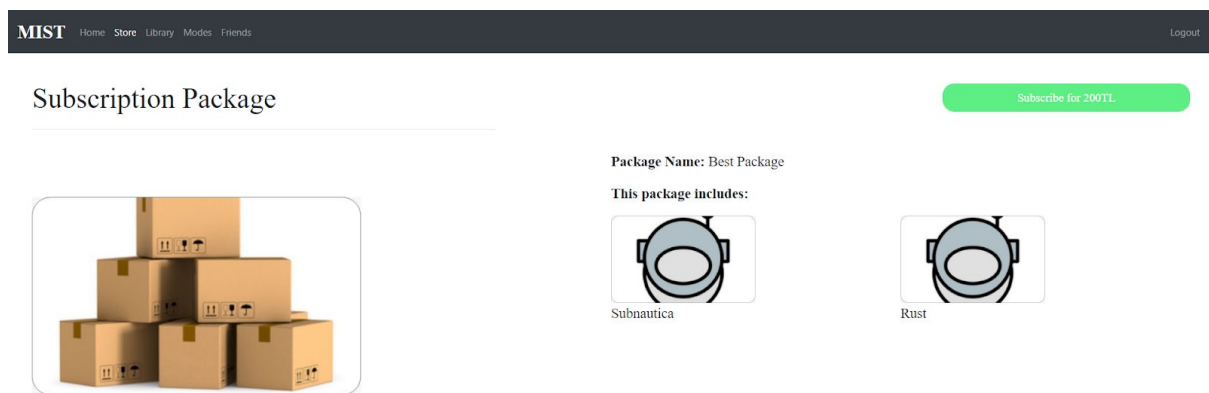


Figure 13: Subscription Package Details

Subscription package details include package name and the games the package contains. By clicking at the upper right corner button, one can subscribe to a package and download the games without buying.

7.2.8. Video Game Details-Not Bought

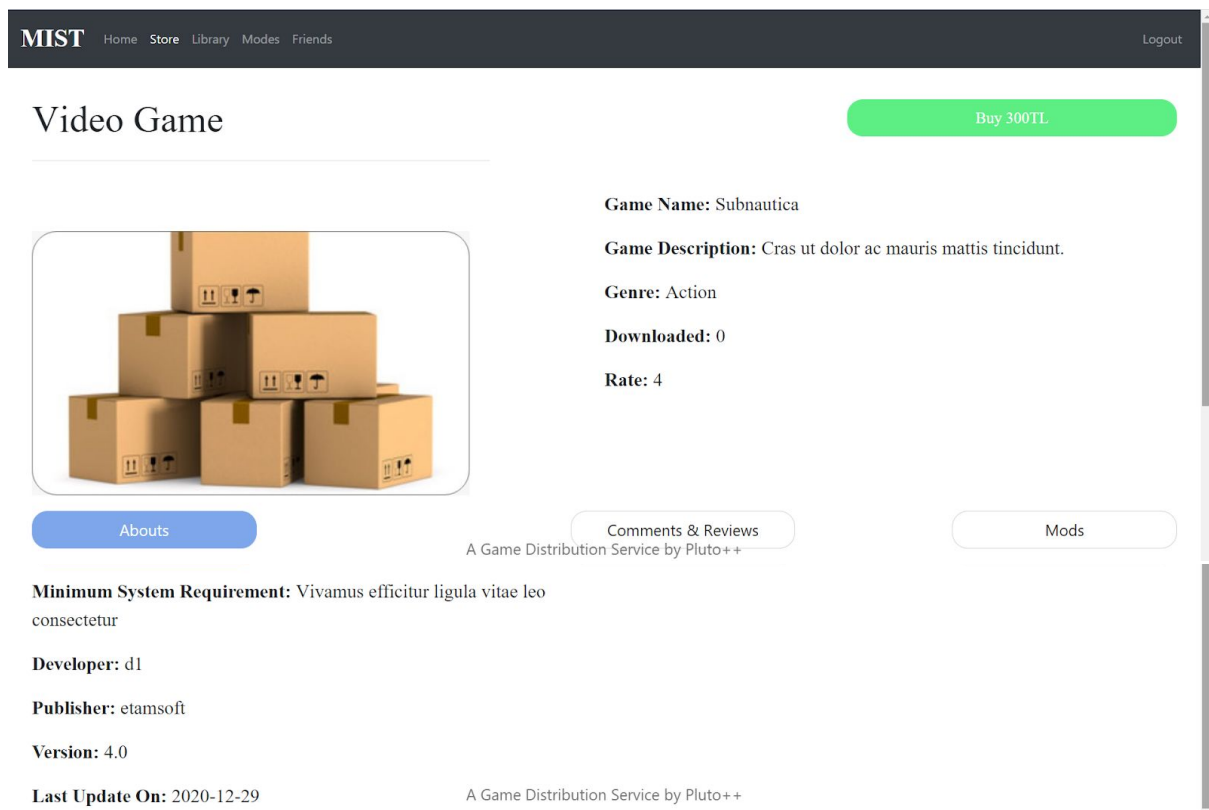


Figure 14: Video Game Details-About Tab

Video game page shows information about a video game as demonstrated in Figure 14. In About Tab, minimum system requirements, publisher and developer of the game, current

version and the last update is shown. If the game is newly published and the developer company did not update the game, the last update shows the publish date.

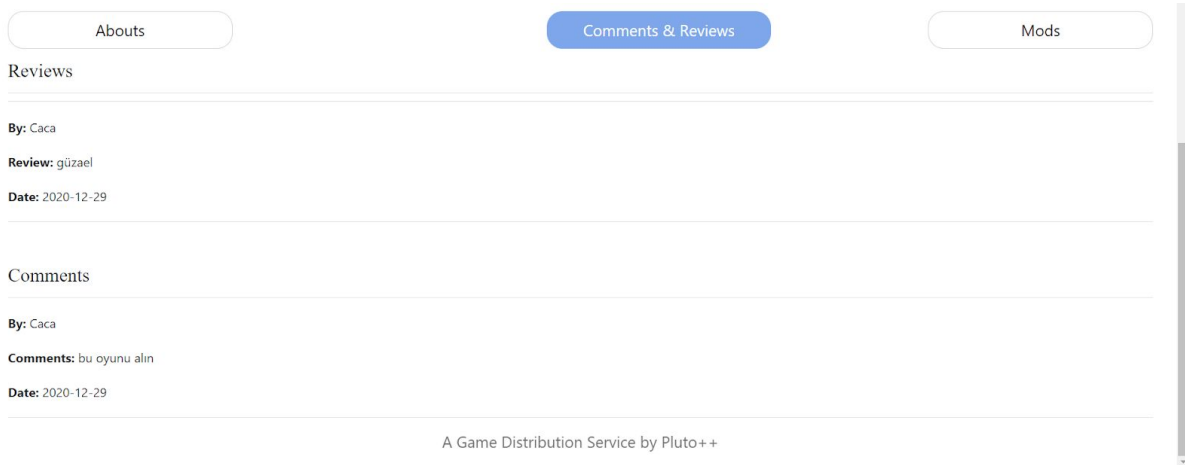


Figure 15: Video Game Details-Comments & Review Tab

In Comments and Reviews Tab, reviews and comments that have been made to the video game are shown. In *Figure 15*, as the user has not bought the game yet, comments or reviews cannot be entered in this case. If the user has installed the game, then the user is able to write a comment to the game. In addition to that, curators are able to write a review of the game if they have installed the game.



Figure 16: Video Game Details-Modes Tab

In the Mods tab, built mods for the game is shown. Because the user had not bought and downloaded the game yet, mods cannot be installed, therefore, install buttons are not available for the case in *Figure 16*.

7.2.9. Video Game Details-Bought

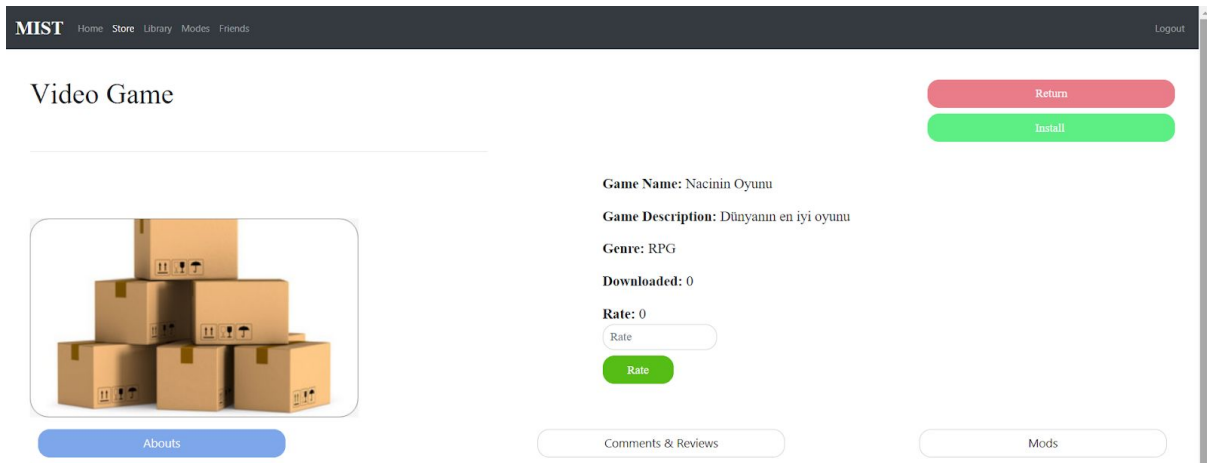


Figure 17: Video Game Details-About Tab After Buying the Game

If the user is currently in about page and still has not installed the game, but bought it, the install and return button is shown at the upper right corner. One can return the game and take back the money or install the game. Moreover, because now the game is bought, the rate option becomes available for the user. The example screen design is shown in Figure 17.

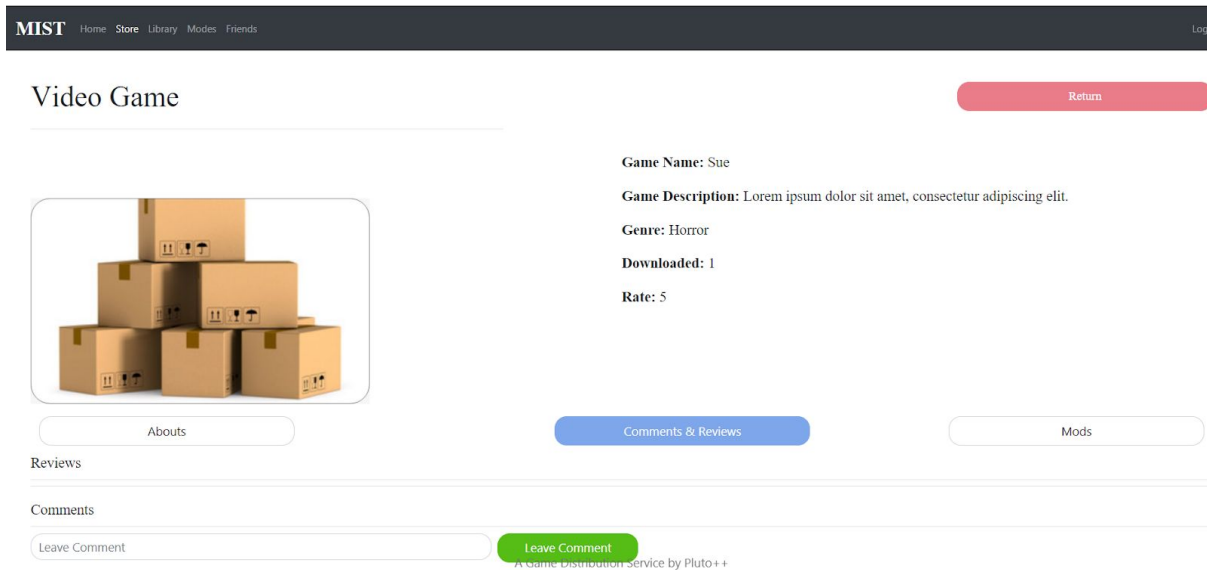


Figure 18: Video Game Details-Comments & Reviews Tab After Buying the Game

If the user currently in comments and review page, leave comment functionality is shown after buying the game. If the user is curator, which will be shown separately in section 6.3, leave review functionality will also be opened.

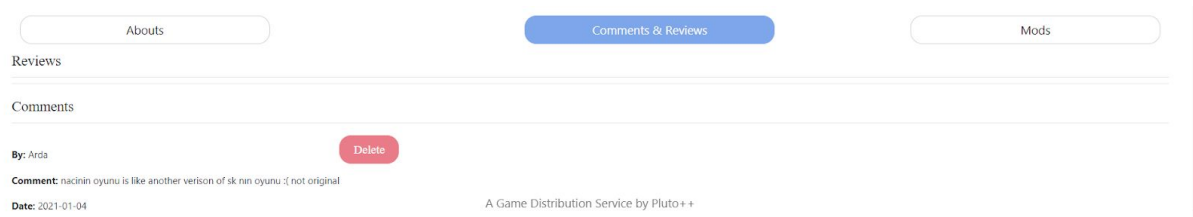


Figure 19: Video Game Details-Comments & Reviews Tab After Leaving a Comment

After leaving a comment, the user can delete the comment if the comment belongs to that user. Moreover, leaving comment operations can only be done once by users, however, if the user deletes the previous comment, he/she can leave a comment again.

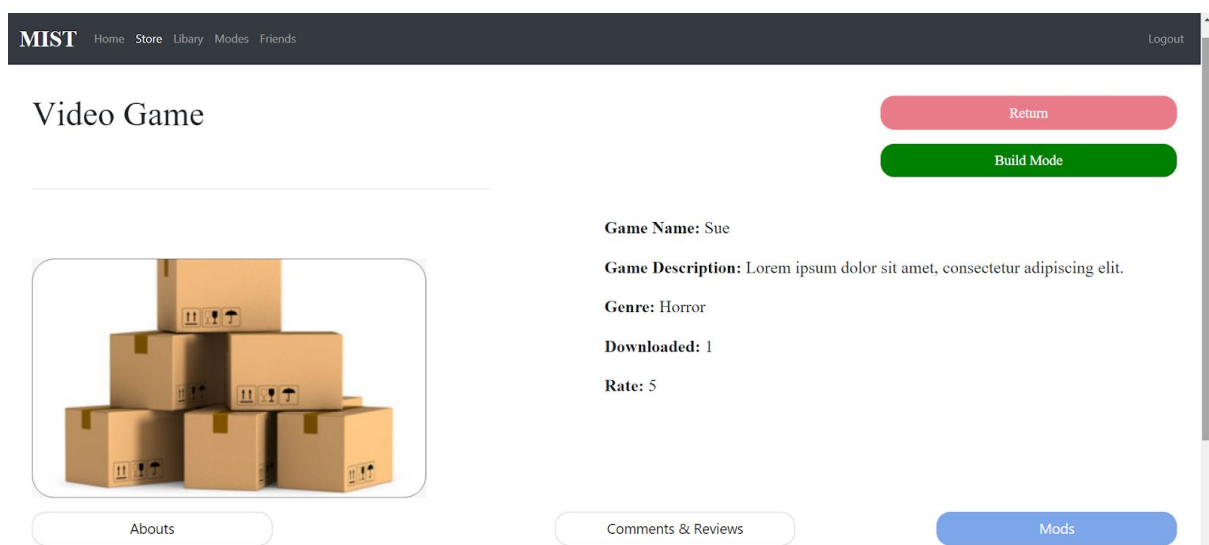


Figure 20: Video Game Details-Mods Tab After Buying the Game

If the user is currently in the Mods page, then, the build mode button is displayed after buying the game. The user can build a mode for that specific game.

7.2.10. Video Game Details-Bought-Installed

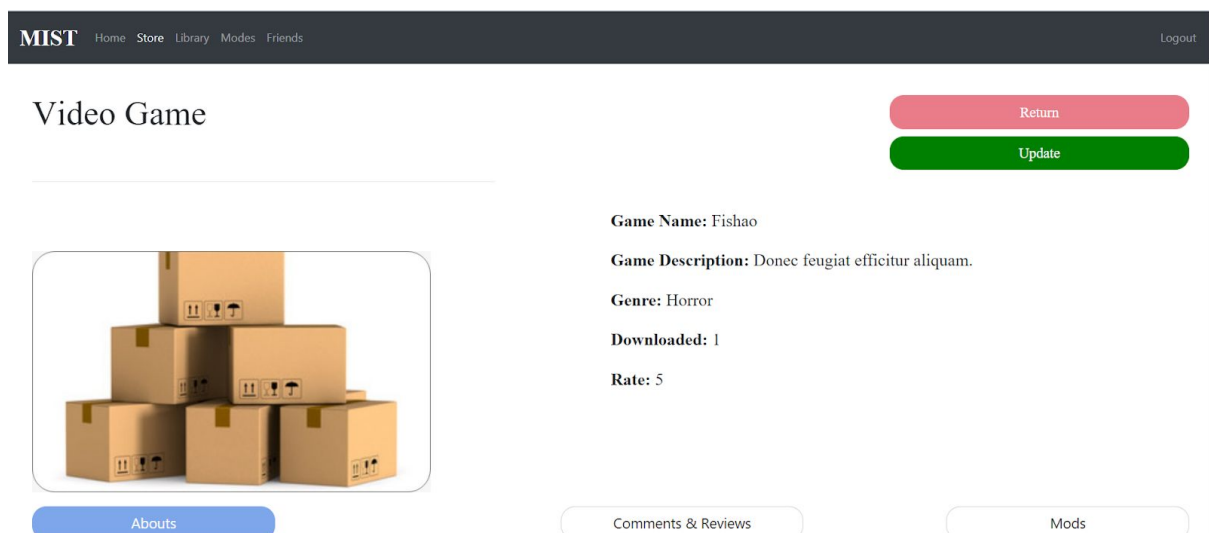


Figure 21: Video Game Details-About Tab after Installing the Game

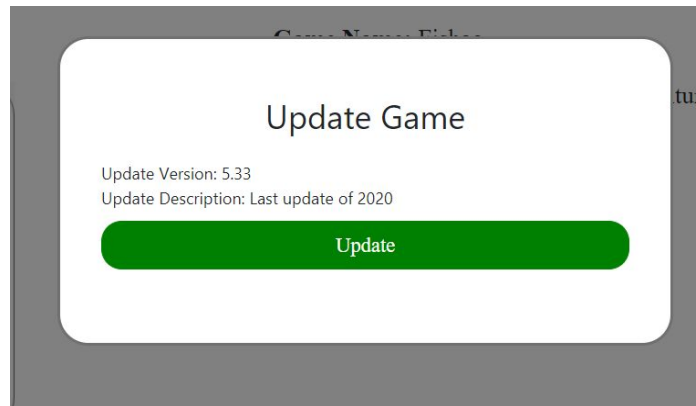


Figure 22: Update Pop-up After Clicking Update Button

After installing the game, the uninstall and return button is displayed as in *Figure 21*. If any update is done by the developer company, the user can update the game seeing information about update attributes as in *Figure 22*.

7.2.11. Build Mode

A screenshot of a web application showing a 'Create Mode' form. The form is white with rounded corners and is centered on a dark gray background. At the top, it says 'Create Mode'. Below that, there are three input fields: 'Name of the Mode', 'Mode Size', and 'Mode Description'. At the bottom, there is a large green button with the text 'Create Mode' in white.

Figure 23: Build Mode Screen

After clicking on the build mode button from the video game mods section, a screen to create the mode is shown as in *Figure 23*. Filling all fields, the user can create the mode for that game by clicking the button. The system will automatically direct the user to the Modes page.

7.2.12. Mode Details

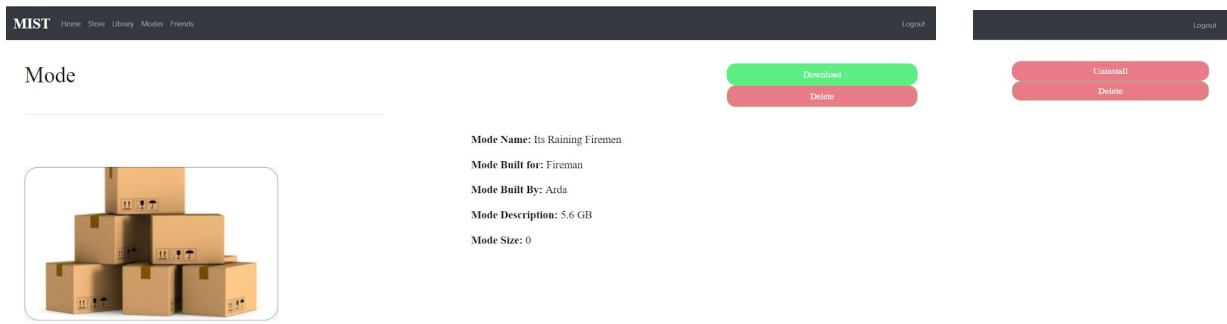


Figure 24: Mode Details

If the user clicks on some specific mode from either the Modes page or Modes section of a video game, a page is shown that displays the details of mode. If the current user is the builder of the mode, then, the delete button is displayed to delete the mode. There are also download-Uninstall buttons.

7.3. Curator Manual

Curator is one type of user, therefore, there are few differences from user described in section 6.2.

7.3.1. Home

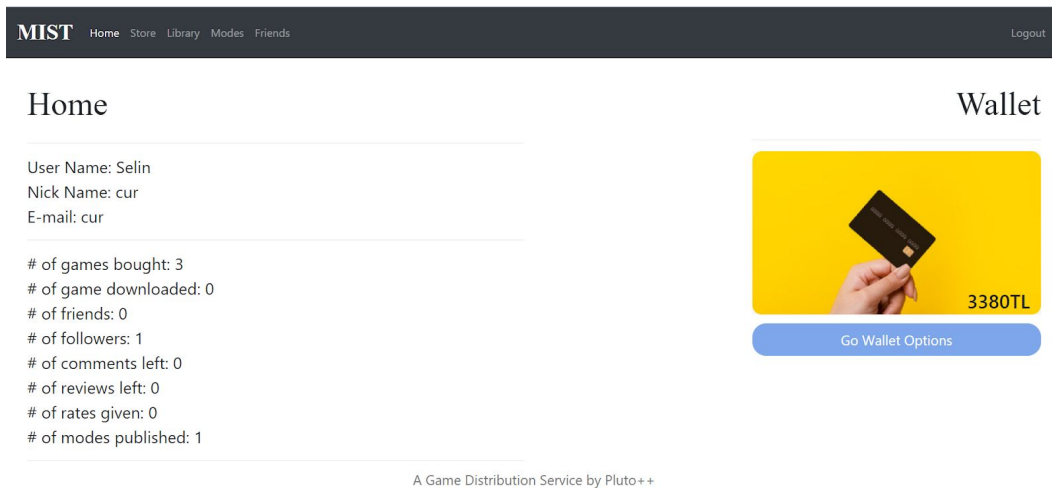


Figure 25: Home Page for Curator

As an extra information besides regular user type, there are “number of followers” and “number of reviews left” sections in the home page of curators.

7.3.2. Video Game Details-Review

Abouts Comments & Reviews Mods

Reviews

Leave Review Leave Review

Comments

Leave Comment Leave Comment

By: Arda

Comments: nacinin oyunu is like another verison of sk nin oyunu :(not original

Date: 2021-01-04

A Game Distribution Service by Pluto++

Figure 26: Video Game Details-Review Tab for Curator

As an additional feature, curators have the functionality to leave a review to games they bought, and their review is shown at the top of the comments and reviews tab.

7.4. Developer Company Manual

7.4.1. Home

MIST Home Games Logout

Home

Company Name: sss

E-mail: ssss

of games approved: 1

of games declined: 0

of requests online: 2

A Game Distribution Service by Pluto++

Figure 27: Home Page for Developer Company

For a developer company, the user interface does not contain comments, reviews, wallet or friends information as developer companies do not have these functionalities. Instead, a number of developer's games approved, declined and current requests are shown.

7.4.2. Games

In the games page of developer company user type, there are several options that a developer company can do.

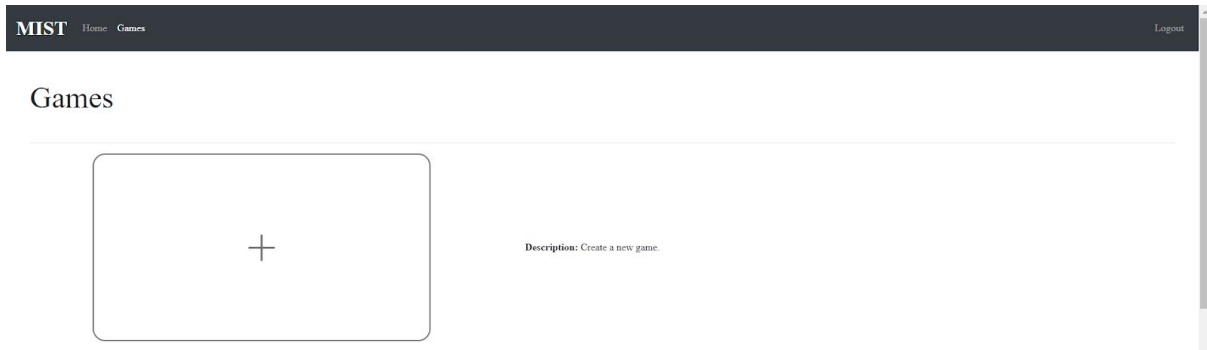


Figure 28: Games Page New Game Section

A developer company can create games by clicking the plus icon in the beginning of the page.

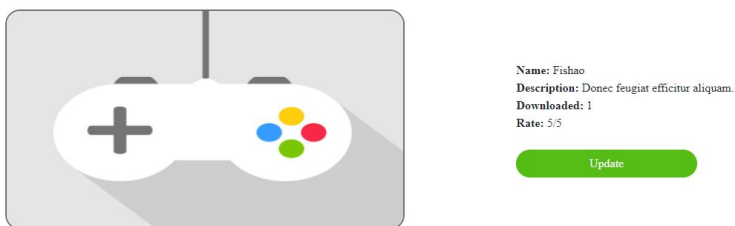


Figure 29: Games Page Update Section

A developer company can update their games that are published on the store.



Figure 30: Games Page Request Section

A developer company can send a publish request to any publisher company by clicking the blue request button.



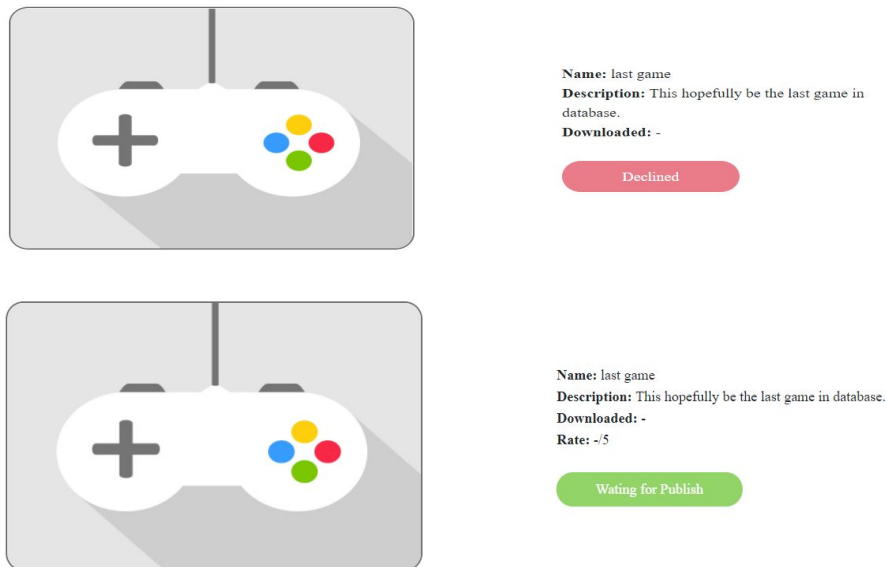


Figure 31: Games Page Approval Status Sections

A developer company can send a new publish request for a game that has been declined by the selected company. Hence, by this, the developer company can both see the declined status from the button and send a new publish request to a publisher company.

7.4.3. Create Game

Figure 32: Create Game Pop-up

When the developer company clicks a new game field, the page will be shown as in *Figure 32*. The developer company will add an image for the game, enter name, genre and version information along with the description and minimum system requirements.

7.4.4. Update Game

A white rounded rectangle form with a grey border. At the top, it says 'Update Game'. Below that, it says 'For the game: Fishao'. There are two input fields: 'New Game Version' and 'Description'. At the bottom, there is a green button labeled 'Update'. Below the form, it says 'A Game Distribution Service by Pluto++'.

Figure 33: Update Game Screen

Developer companies can update their own game after clicking the Update button. A new screen will be shown as in Figure 33. The developer company

A white rounded rectangle form with a grey border. At the top, it says 'Send Request'. Below that, it says 'For the game: last game'. There is a label 'Choose a Publisher Company:' followed by a dropdown menu with the text 'Select a Publisher Company'. At the bottom right, there is a green button labeled 'Send'.

will enter a new version and the update description. The date of update will be automatically obtained from the system.

7.4.5. Request

Figure 34: Request Screen

The developer company sends a request to publisher companies for publishing their game. The publisher company is chosen from the drop down field as can be seen in Figure 34.

7.5. Publisher Company Manual

7.5.1. Home



Figure 35: Home Page for Publisher Company

For a publisher company, the user interface for the home page is similar to Developer Company as shown in *Figure 35*. However, the number of approved and declined games and requests are the publisher's decisions.

7.5.2. Requests

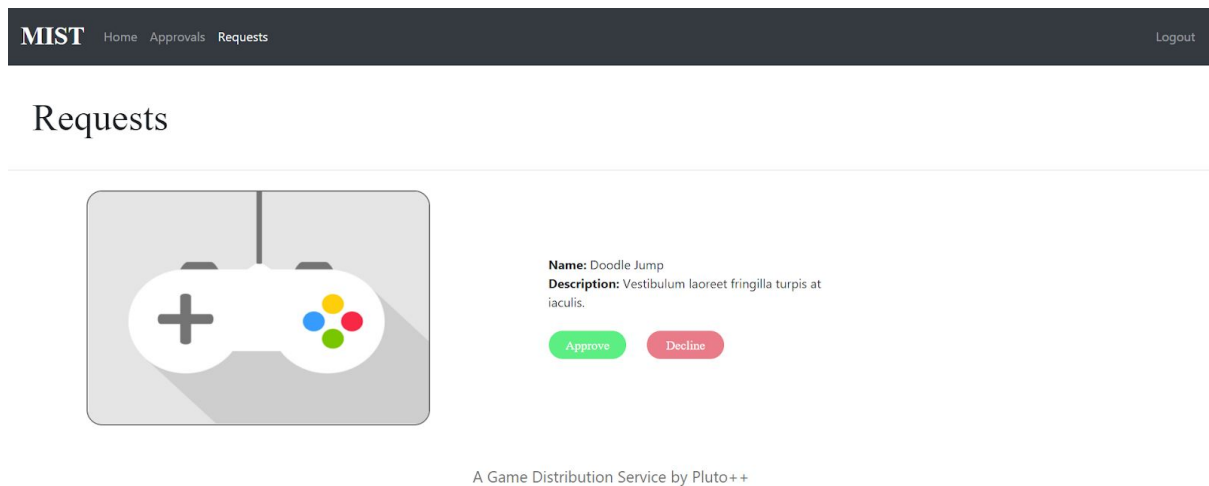


Figure 36: Request Page

The publisher company type has a requests page that they can see the game requests came for them to publish. The publisher company either approves or declines the requests from this page as demonstrated in *Figure 36*.

7.5.3. Approvals

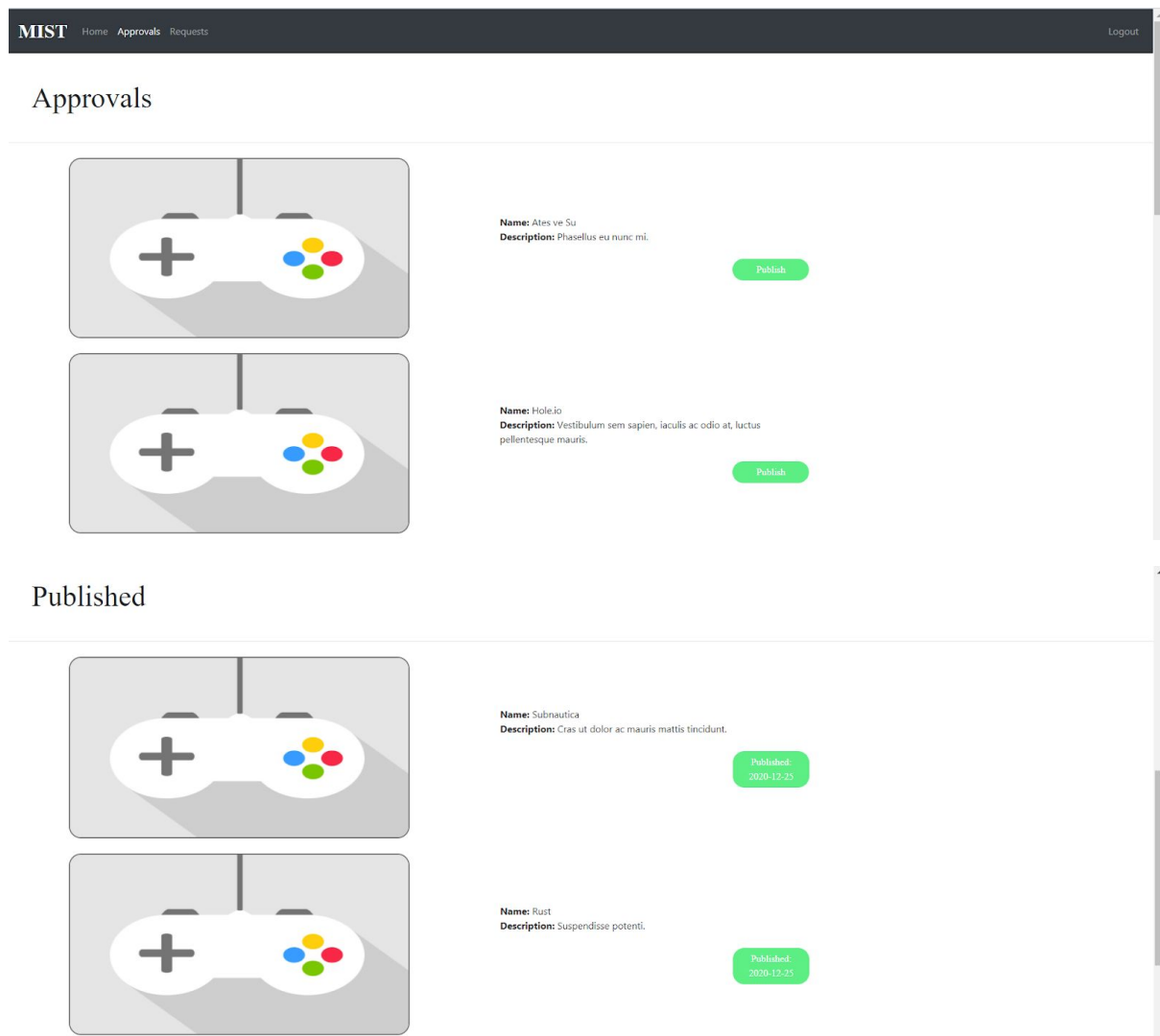


Figure 37: Approval Page

Publisher companies have an approval page to see the games they have approved before. From this page, they can see the information about the game and the status of whether they have published or not. If they have not published yet, the publish button is used. After clicking the publish button from the Approvals page, the publisher company sets the price of the game and then publishes as in *Figure 37*.

8. Website

The website of the project along with report and the repository that holds the source code can be found following:

<https://subfly.github.io/Stream-Site/#/>