



**İHSAN DOĞRAMACI BİLKENT ÜNİVERSİTY**

**SPRING 2018**

**CS353 - DATABASE SYSTEMS**

**TERM PROJECT FINAL REPORT**

**Social Gaming Marketplace - Ethereum**

**GROUP 23**

**GROUP MEMBERS:**

1. Bikem Çamli, 21501492, Section 2
2. Mert Osman Dönmezyürek, 21301890, Section 2
3. Nursena Kal, 21501322, Section 1
4. Mahin Khankishizade, 21503495, Section 1

## **Table of Contents**

<b>1.Introduction</b>	<b>5</b>
1.1 Revised E/R Model	6
1.2 The Changes Made in ER Diagram	6
<b>2. Relation Schema</b>	<b>7</b>
2.1 Game	7
2.2 Game_Category	7
2.3 Company	7
2.4 Event	8
2.5 Player	8
2.6 Premium_Player	9
2.7 Admin	9
2.8 Information	9
2.9 Message	10
2.10 Comment	10
2.11 Company_Info	10
2.12 Updates	11
2.13 Discount	11
2.15 Bought	12
2.16 Played	13
2.17 Chats	13
2.18 Blocked	14
2.19 Friended	14
2.20 Banned	14
2.21 Warned	15
2.22 Approves	15
2.23 Likes	16
2.24 Multiplayed	16
<b>3. Implementation</b>	<b>17</b>
<b>4. Sample Output Reports</b>	<b>18</b>
<b>5.User Manual</b>	<b>34</b>
5.1 Index Page	34
5.2 Admin Login Page	34
5.3 Admin Marketplace	35
5.4 Admin Profile	36

5.5 Admin Manage Companies	37
5.6 Admin Manage Players	38
5.7 Admin Manage Comments	39
5.8 Admin Manage Categories	40
5.9 Admin Account Settings	41
5.10 Company Sign Up Page	42
5.11 Company Login Page	43
5.12 Company Marketplace	45
5.13 Company Event Addition	45
5.14 Company Event Management	45
5.15 Company Games	46
5.16 Company Game Addition	46
5.17 Company Update Game	47
5.18 Company Manages Game	48
5.19 Company Game Statistics	49
5.20 Company Profile Information	50
5.21 Company Account Settings	51
5.22 Player Sign Up Page	52
5.23 Player Login Page	54
5.24 Player Marketplace	55
5.25 Player Games	56
5.26 Player Profile	56
5.27 Friend Profile Information	57
5.28 Friend Profile History	58
5.29 Player Profile Information	59
5.30 Player Game History	60
5.31 Add Funds to Wallet	61
5.32 Player Account Settings	62
5.33 Chat pop-up	63
5.34 Premium Player Profile	65
5.35 Player Writes Comment	65
<b>6. Advanced Database Components</b>	<b>66</b>
6.1 Views	66
6.2 Constraints	66
6.3 Triggers	67
6.4 Stored Procedures	67



## **1. Introduction**

Ethereal is a web-based social platform where the users can buy and play games from the game store. In addition, by using this platform, the users are able to interact with the fellow users either in terms of the game or in terms of the communication.

The players are able to see friends, play games, comment on games. Companies are able to upload their games, see the statistics of games they own, create events and apply discounts accordingly. Admins are able to manage every event, comment, game, company and player that have been added to the system.

Main entity of this project is games. Each game will have developers, publisher, description, published date and a picture stored in the database. In addition, a game will have a game category, which will limit the players to an age limit. In addition, there is a company entity which owns a game. Therefore, they will be able to add new games and provide events and with these events, there will be discounts on games.

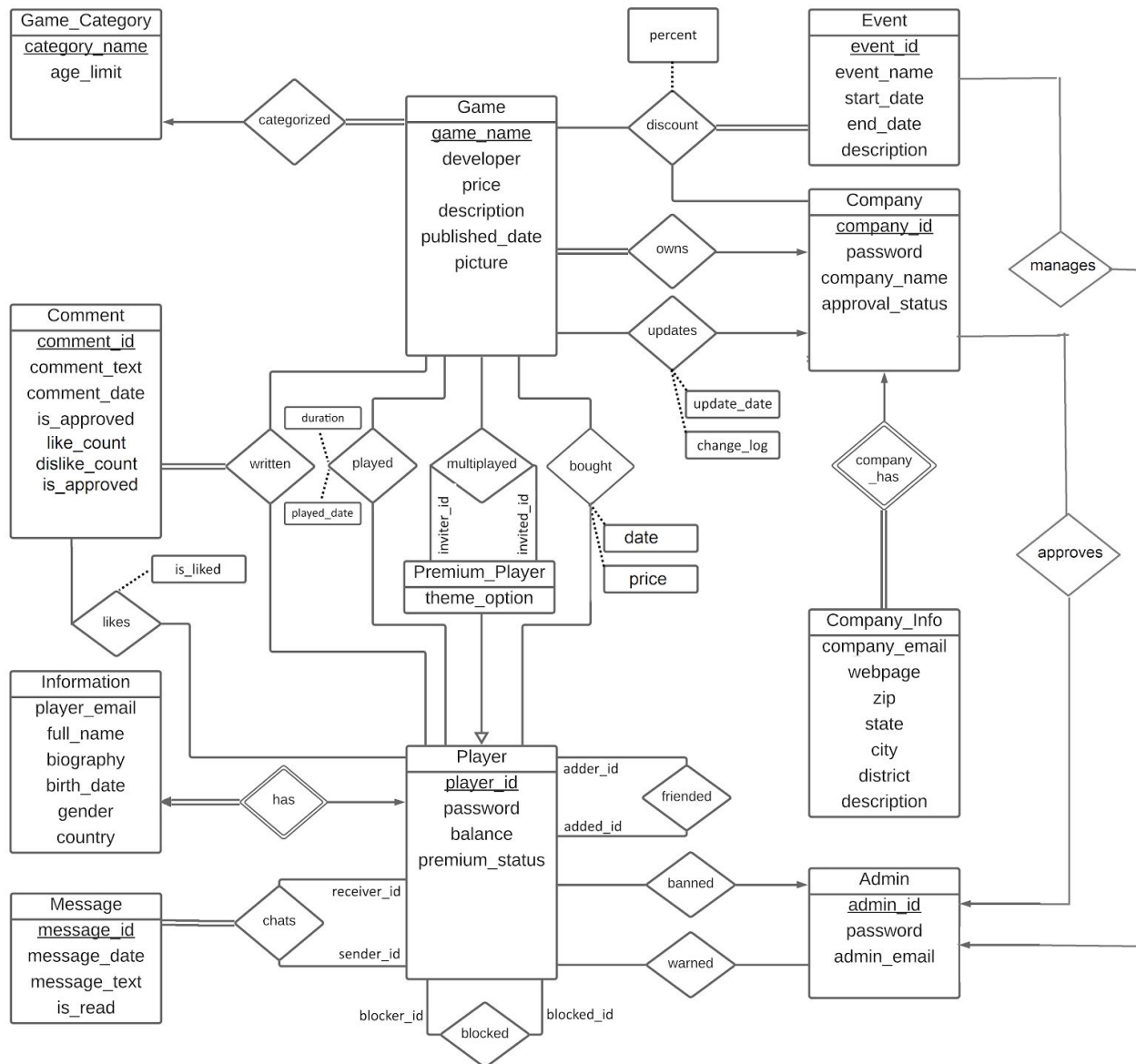
Every member of our group has worked on different parts of this system. We worked on code together, therefore every member have worked on every part of this project. Here is the main work division among the team :

Mahin have developed Player, some of Company related pages and Mert developed Company and main UI of the system and created databases.

Nursena and Bikem have developed Company and Admin related pages, also worked on database.

Using different types of team communication systems helped us on dividing work equally and improving our hard and soft skills. We have used Trello during implementation stage of our project.

## 1.1 Revised E/R Model



## 1.2 The Changes Made in ER Diagram

- A relation between Admin and Event is added, named “manages”.
- A relation between Admin and Company is added, named “approves”.
- New attributes added to bought relation, named “date” and “price”.

- New attributes added to Comment table named “like\_count” , “dislike\_count” and “is\_approved”.
- Relations between Information and Player, Company\_Info and Company has changed to one to one.

## **2. Relation Schema**

### **2.1 Game**

#### **Relational Model:**

Game( game\_name, developer, description, published\_date, picture, price, company\_id, category\_name)

#### **Functional Dependencies:**

game\_name -> developer, description, published\_date, picture, price, company\_id, category\_name

#### **Keys**

**Candidate Keys:** { (game\_name) }

**Primary key:** game\_name

**Foreign Key:** None

**Normal Forms:** BCNF

### **2.2 Game\_Category**

#### **Relational Model:**

Game\_Category( category\_name, age\_limit)

#### **Functional Dependencies:**

category\_name -> age\_limit

#### **Keys**

**Candidate Keys:** { (category\_name) }

**Primary key:** category\_name

**Foreign Key:** None

**Normal Forms:** BCNF

### **2.3 Company**

#### **Relational Model:**

Company( company\_id, company\_name, password, is\_approved)

**Functional Dependencies:**

company\_id -> company\_name, password, is\_approved

**Keys**

**Candidate Keys:** { (company\_id) }, { (company\_name) }

**Primary key:** company\_id

**Foreign Key:** None

**Normal Forms:** BCNF

**2.4 Event**

**Relational Model:**

Event( event\_id, event\_name, start\_date, end\_date, description)

**Functional Dependencies:**

event\_id -> event\_name, start\_date, end\_date, description

**Keys**

**Candidate Keys:** { (event\_id) }

**Primary key:** event\_id

**Foreign Key:** None

**Normal Forms:** BCNF

**2.5 Player**

**Relational Model:**

Player( player\_id, password, balance, premium\_status)

**Functional Dependencies:**

player\_id -> password, balance, premium\_status

**Keys**

**Candidate Keys:** { (player\_id) }

**Primary key:** player\_id

**Foreign Key:** None



**Normal Forms:** BCNF

## **2.6 Premium\_Player**

**Relational Model:**

Premium\_Player( player\_id, theme\_option)

**Functional Dependencies:**

player\_id -> theme\_option

**Keys**

**Candidate Keys:** { (player\_id) }

**Primary key:** player\_id

**Foreign Key:** None

**Normal Forms:** BCNF

## **2.7 Admin**

**Relational Model:**

Admin( admin\_id, password, admin\_email)

**Functional Dependencies:**

admin\_id -> password, admin\_email

**Keys**

**Candidate Keys:** { (admin\_id) }, { (admin\_email) }

**Primary key:** admin\_id

**Foreign Key:** None

**Normal Forms:** BCNF

## **2.8 Information**

**Relational Model:**

Information( player\_id, player\_email, full\_name, birth\_date, gender, country, biography)

FK: player\_id references Player(player\_id)

**Functional Dependencies:**

player\_id, player\_email -> full\_name, birth\_date, gender, country, biography

**Keys**

**Candidate Keys:** { (player\_id, player\_email) }

**Primary key:** player\_id, player\_email

**Foreign Key:** player\_id references Player

**Normal Forms:** BCNF

## 2.9 Message

**Relational Model:**

Message( message\_id, message\_date, message\_text)

**Functional Dependencies:**

message\_id -> message\_date, message\_text

**Keys**

**Candidate Keys:** { (message\_id) }

**Primary key:** message\_id

**Foreign Key:** None

**Normal Forms:** BCNF

## 2.10 Comment

**Relational Model:**

Comment( comment\_id, comment\_text, comment\_date, like\_count, dislike\_count, is\_approved)

**Functional Dependencies:**

comment\_id -> comment\_text, comment\_date, like\_count, dislike\_count, is\_approved

**Keys:**

**Candidate Keys:** { (comment\_id) }

**Primary key:** comment\_id

**Foreign Key:** None

**Normal Forms:** BCNF

## 2.11 Company\_Info

**Relational Model:**

Company\_Info( company\_id, company\_email, webpage, zip, state, city, district, description, )

FK: company\_id references Company(company\_id)

**Functional Dependencies:**

company\_id, company\_email -> webpage, zip, state, city, district, description, description  
zip -> state, city, district

**Keys:**

**Candidate Keys:** { (company\_id, company\_email) }

**Primary key:** company\_id, company\_email

**Foreign Key:** company\_id references Company

**Normal Forms:** BCNF

**2.12 Updates****Relational Model:**

Updates( game\_name, company\_id, update\_date, change\_log)

FK: company\_id references Company( company\_id)

FK: game\_name references Game(game\_name)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (game\_name, company\_id) }

**Primary key:** game\_name, company\_id

**Foreign Key:** game\_name references Game, company\_id references Company

**Normal Forms:** BCNF

**2.13 Discount****Relational Model:**

Discount ( game\_name, company\_id, event\_id, percent)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (game\_name, company\_id, event\_id) }

**Primary key:** game\_name, company\_id, event\_id

**Foreign Key:** game\_name references Game,

company\_id references Company,  
event\_id references Event

**Normal Forms:** BCNF

## 2.14 Written

### Relational Model:

Written( game\_name, player\_id, comment\_id )

FK: game\_name references Game(game\_name)

FK: player\_id references Player(player\_id)

FK: comment\_id references Comment(comment\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (game\_name, player\_id, comment\_id) }

**Primary key:** game\_name, player\_id, comment\_id

**Foreign Key:** game\_name references Game,  
comment\_id references Comment,  
player\_id references Player

**Normal Forms:** BCNF

## 2.15 Bought

### Relational Model:

Bought( game\_name, player\_id, price, date )

FK: game\_name references Game(game\_name)

FK: player\_id references Player(player\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (game\_name, player\_id) }

**Primary key:** game\_name, player\_id

**Foreign Key:** game\_name references Game,  
player\_id references Player

**Normal Forms:** BCNF

## 2.16 Played

### Relational Model:

Played( game\_name, player\_id, duration, played\_date)  
FK: game\_name references Game(game\_name)  
FK: player\_id references Player(player\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (game\_name, player\_id) }

**Primary key:** game\_name, player\_id

**Foreign Key:** game\_name references Game,  
player\_id references Player

**Normal Forms:** BCNF

## 2.17 Chats

### Relational Model:

Chats( sender\_id, receiver\_id, message\_id )  
FK: sender\_id references Player(player\_id)  
FK: receiver\_id references Player(player\_id)  
FK: message\_id references Message(message\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (sender\_id, receiver\_id, message\_id) }

**Primary key:** sender\_id, receiver\_id, message\_id

**Foreign Key:** sender\_id reference to Player,  
receiver\_id reference to Player,  
message\_id reference to Message

**Normal Forms:** BCNF

## 2.18 Blocked

### Relational Model:

Blocked( blocker\_id, blocked\_id )

FK: blocker\_id references Player(player\_id)

FK: blocked\_id references Player(player\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (blocker\_id, blocked\_id) }

**Primary key:** blocker\_id, blocked\_id

**Foreign Key:** blocker\_id reference to Player,  
blocked\_id reference to Player

**Normal Forms:** BCNF

## 2.19 Friended

### Relational Model:

Friended( adder\_id, added\_id )

FK: adder\_id references Player(player\_id)

FK: added\_id references Player(player\_id)

### Functional Dependencies:

None

### Keys

**Candidate Keys:** { (adder\_id, added\_id) }

**Primary key:** adder\_id, added\_id

**Foreign Key:** added\_id references Player,  
adder\_id references Player

**Normal Forms:** BCNF

## 2.20 Banned

### Relational Model:

Banned( player\_id, admin\_id )

FK: player\_id references Player(player\_id)

FK: admin\_id references Admin(admin\_id)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (player\_id, admin\_id) }

**Primary key:** player\_id, admin\_id

**Foreign Key:** player\_id reference to Player,  
admin\_id reference to Admin

**Normal Forms:** BCNF

**2.21 Warned**

**Relational Model:**

Warned( player\_id, admin\_id )

FK: player\_id references Player(player\_id)

FK: admin\_id references Admin(admin\_id)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (player\_id, admin\_id) }

**Primary key:** player\_id, admin\_id

**Foreign Key:** player\_id references Player,  
admin\_id references Admin

**Normal Forms:** BCNF

**2.22 Approves**

**Relational Model:**

Approves( company\_id, admin\_id, status )

FK: company\_id references Company(company\_id)

FK: admin\_id references Admin(admin\_id)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (company\_id, admin\_id) }

**Primary key:** company\_id, admin\_id

**Foreign Key:** company\_id references Company,  
admin\_id references Admin

**Normal Forms:** BCNF

## 2.23 Likes

**Relational Model:**

Likes( player\_id, comment\_id, is\_liked)

FK: player\_id references Player(player\_id)

FK: comment\_id references Comment(comment\_id)

**Functional Dependencies:**

None

**Keys**

**Candidate Keys:** { (player\_id, comment\_id) }

**Primary key:** player\_id, comment\_id

**Foreign Key:** player\_id references Player,  
comment\_id references Comment

**Normal Forms:** BCNF

## 2.24 Multiplayed

**Relational Model:**

Multiplayed( inviter\_id, invited\_id, game\_name)

FK: invited\_id references Player(player\_id)

FK: inviter\_id references Player(player\_id)

FK: game\_name references Game(game\_name)

**Functional Dependencies:**

Nope

**Keys**

**Candidate Keys** { ( (inviter\_id, invited\_id, game\_name) ) }

**Primary key:** inviter\_id, invited\_id, game\_name

**Foreign Key:** inviter\_id references Player,  
invited\_id references Player,



game\_name references Game

**Normal Forms:** BCNF

### 3. Implementation

- We created a temporary MySQL server and the Ethereum database on the server4now.net website. We used PDO on PHP while using our SQL statements.
- We used Chrome Database Admin to manage our database.
- To see our pages we used dijkstra.ug.bcc.bilkent.edu.tr server's public\_html directory.
- We implemented our back-end code in PHP .
- We implemented our front-end in HTML, CSS and JS.

Checking Pages Process:

1. Open FileZilla
2. Connect to dijkstra.ug.bcc.bilkent.edu.tr
3. Open public\_html directory
4. Upload written codes to the directory

Database Connection Process:

1. Open Chrome Database Admin
2. Enter host name: mysql7002.site4now.net
3. Enter password
4. Connect

Database Connection Inside PHP:

1. We created database.php file which connects to database from the start of the website
2. We included 'require database.php' code segment in each php file

Additional Information

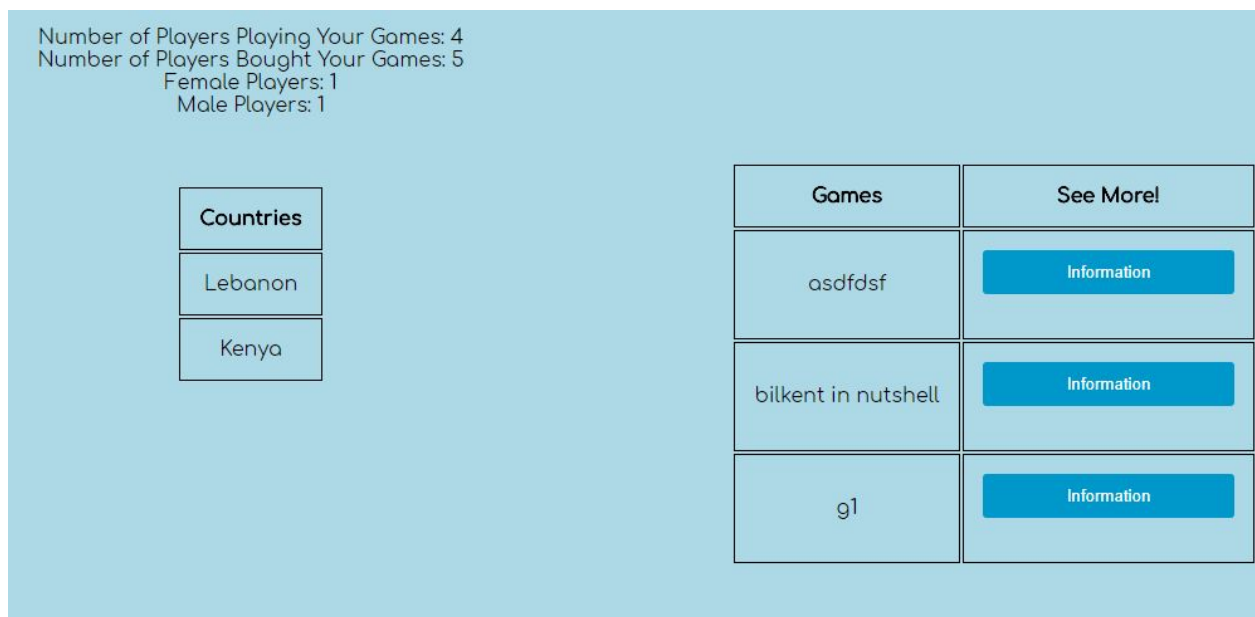
1. There are some php files that have no GUI (for example, buy\_game.php). Those kind of files just proceed a function. For passing variables to those files, we used GET methods.
2. For updating information, account settings, events or games we used POST methods.

Problems Occurred During Implementation

1. During implementing game addition page, there was a problem which occurred because of one extra space that was on value giving part on PHP echo statement. According to the Google, this problem could only happen because of database but we were sure that database was implemented correctly. We had to check every statement that we have written so far and we noticed the one extra space after five hours.
2. Inserting a row to a table using PDO had a different way of accessing variables that are sent by POST method. Before knowing this we had problems with inserting a row, then we learned that using an array inside a execute function call.

3. We tried to change Discount table before changing Event table. But it did not work. Then we realized that Discount table is only a relation so we changed the order of each changing place.
4. Deleting row from Game table was a problem because we also had to delete all rows from all relations that are bounded to Game table. We solved this problem with adding a trigger to our database.

#### 4. Sample Output Reports



Pic 1. Company Overall Statistics

In Pic 1. There are 6 reports.

1. Number of players playing the a company's games

```
<?php
    $records = $conn->prepare('select count(distinct player_id) as number from played natural join game where company_id = :company_id');
    $records->bindParam(':company_id', $_SESSION['company_id']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);
?>
<label>Number of Players Playing Your Games: </label>
<label><?php echo $results['number'];?></label><br />
```

2. Number of players bought the company's games

```

<?php
    $records = $conn->prepare('select count(distinct player_id) as number from bought where game_name in (select game_name from game where company_id =:company_id)');
    $records->bindParam(':company_id', $_SESSION['company_id']);
    $records->execute();
    $results = $records->fetch(PDO::FETCH_ASSOC);
>
<label>Number of Players Bought Your Games: </label>
<label><?php echo $results['number'];?></label><br />

```

### 3. Number of female players playing a company's games

```

<?php

```

```

    $records = $conn->prepare('select count(distinct player_id) as females from information
natural join played where gender = \'female\' and game_name in (select game_name from game
where company_id =:company_id)');

```

```

    $records->bindParam(':company_id', $_SESSION['company_id']);

```

```

    $records->execute();

```

```

    $results = $records->fetch(PDO::FETCH_ASSOC);

```

```

?>

```

```

<label>Female Players: </label>

```

```

<label><?php echo $results['females'];?></label><br />

```

### 4. Number of male players playing a company's games

```

<?php

```

```

    $records = $conn->prepare('select count(distinct player_id) as males from information natural
join played where gender = \'male\' and game_name in (select game_name from game where
company_id =:company_id)');

```

```

    $records->bindParam(':company_id', $_SESSION['company_id']);

```

```

    $records->execute();

```

```

    $results = $records->fetch(PDO::FETCH_ASSOC);

```

```

?>

```

```

<label>Male Players: </label>

```

```

<label><?php echo $results['males'];?></label><br />

```

### 5. Countries where a company's games are played

```

<table align = "center">
<thead>
<th>Countries</th><br />
</thead>
<tbody>
<?php
$records = $conn->prepare('SELECT distinct country from information natural join played WHERE game_name in (select game_name from game where company_id =:company_id)');
$records->bindParam(':company_id', $_SESSION['company_id']);
$records->execute();
$results = $records->fetchAll();

foreach($results as $result)
{
echo "<tr>";
echo "<td>" . $result['country'] . "</td>" . "<br>";
echo "</tr>";
}
?>
</tbody>
</table>

```

Number of Players Playing Your Games: 4	
Number of Players Bought Your Games: 5	
Female Players: 1	
Male Players: 1	

Countries
Lebanon
Kenya

Game Name	Age	Count of Players
g1	10	2

Pic 2. Company Statistics for Each Game

There are 6 reports in Pic 2. This page is for a particular game of a particular company.

1. Number of players playing a game

```
<?php
```

```

$records = $conn->prepare('select count(distinct player_id) as number from played where
game_name = :game_name');

```

```

$records->bindParam(':game_name', $game_name);

```

```

$records->execute();

```

```

$results = $records->fetch(PDO::FETCH_ASSOC);

```

```
?>
```

```
<label>Number of Players Playing Your Games: </label>
```

```
<label><?php echo $results['number'];?></label><br />
```

## 2. Number of players bought a game

<?php

```
$records = $conn->prepare('select count(distinct player_id) as number from bought where  
game_name = :game_name');
```

```
$records->bindParam(':game_name', $game_name);
```

```
$records->execute();
```

```
$results = $records->fetch(PDO::FETCH_ASSOC);
```

?>

<label>Number of Players Bought Your Games: </label>

<label><?php echo \$results['number'];?></label><br />

## 3. Number of female players playing a game

<?php

```
$records = $conn->prepare('select count(distinct player_id) as females from information  
natural join played where gender = \'female\' and game_name = :game_name');
```

```
$records->bindParam(':game_name', $game_name);
```

```
$records->execute();
```

```
$results = $records->fetch(PDO::FETCH_ASSOC);
```

?>

<label>Female Players: </label>

<label><?php echo \$results['females'];?></label><br />

## 4. Number of male players playing a game

<?php

```
$records = $conn->prepare('select count(distinct player_id) as males from information natural  
join played where gender = \'male\' and game_name = :game_name');
```

```
$records->bindParam(':game_name', $game_name);
```

```
$records->execute();
```

```
$results = $records->fetch(PDO::FETCH_ASSOC);
```

?>

<label>Male Players: </label>

<label><?php echo \$results['males'];?></label><br />

5. Countries that a game is being played in

<table align = "center">

<thead>

<th>Countries</th><br />

</thead>

<tbody>

<?php

\$records = \$conn->prepare('SELECT distinct country from information natural join played  
WHERE game\_name = :game\_name');

\$records->bindParam(':game\_name', \$game\_name);

\$records->execute();

\$results = \$records->fetchAll();

foreach(\$results as \$result)

{

echo "<tr>";

echo "<td>" . \$result['country'] . "</td>" . "<br>";

echo "</tr>";

}

?>

</tbody>

</table>

6. The age and number of players playing a game

<table align = "right">

```

<thead>
    <th>Game Name</th><br />
    <th>Age</th><br />
    <th>Count of Players</th><br />
</thead>
<tbody>
    <?php

```

```

    $records = $conn->prepare('SELECT game_name, age, number from (SELECT game_name,
(played_date - birth_date) as age, count(player_id) as number from Player natural join
Information natural join Played group by game_name) as T where game_name = :game_name');
    $records->bindParam(':game_name', $game_name);
    $records->execute();
    $results = $records->fetchAll();

```

```

foreach($results as $result)
{
    echo "<tr>";
    echo "<td>" . $result['game_name'] . "</td>" . "<br>";
    echo "<td>" . $result['age'] . "</td>" . "<br>";
    echo "<td>" . $result['number'] . "</td>" . "<br>";
    echo "</tr>";
}
?>
</tbody>
</table>

```

Event	Start Date	End Date
1233	2018-05-02	2018-05-26
hi	1990-05-05	2018-02-03

Pic 3. Company Event History

Companies can see their event history.

```
<table align = "right">
```

```
<thead>
```

```
<th>Event</th><br />
```

```
<th>Start Date</th><br />
```

```
<th>End Date</th><br />
```

```
</thead>
```

```
<tbody>
```

```
<?php
```

```
$records1 = $conn->prepare('SELECT distinct event_id, start_date, end_date from discount
natural join event where company_id = :company_id');
```

```
$records1->bindParam(':company_id', $_SESSION['company_id']);
```

```
$records1->execute();
```

```
$results1 = $records1->fetchAll();
```

```
foreach($results1 as $result)
```

```
{
```

```
    echo "<tr>";
```

```
    echo "<td>" . $result['event_id'] . "</td>" . "<br>";
```

```
    echo "<td>" . $result['start_date'] . "</td>" . "<br>";
```



```

        echo "<td>" . $result['end_date'] . "</td>" . "<br>";
        echo "</tr>";
    }
    ?>
</tbody>
</table>

```

Games	Developer	Description	Release Date	Category	Company	Actual Price
g1	d1	really fun game	2000-01-11	romance	a	50

Shopping Details

Price When You Bought	Date When You Bought
50	2018-05-12

Pic 4. Player's Game Shopping Details

Friends	Profiles
p4	See
p6	See
p1	See
p2	See

Pic 5. Player's Friend List

Players can see their friend lists (not only added but also the adders).

```
<table align = "right">
```

```
<thead>
```

```
<th>Friends</th><br />
```

```
<th>Profiles</th><br />
```

```
</thead>
```

```
<tbody>
```

```
<?php
```

```

$records1 = $conn->prepare('SELECT added_id from friended where adder_id = :player_id
union select adder_id from friended where added_id = :player_id ');

```

```
$records1->bindParam(':player_id', $_SESSION['user_id']);
```

```
$records1->execute();
```

```

$results1 = $records1->fetchAll();

foreach($results1 as $result)
{
    echo "<tr>";
    echo "<td>" . $result['added_id'] . "</td>" . "<br>";
    echo "<td><a href='./player_profile_sideView.php?added_id=" . $result['added_id'].
\"><input type='submit' value='See' /></form></td>";

    echo "</tr>";
}
?>
</tbody>
</table>

```

Games	Duration	Date
g1	120	2018-05-11
g2	120	2018-05-11
g3	120	2018-05-11

Pic 6. Player's Played Games List

Players can see their played game history.

```

</table>
<div>
<table align = "center">
<p> Played Games</p>
<thead>

```

```

        <th>Games</th><br />
        <th>Duration</th><br />
        <th>Date</th><br />
    </thead>
    <tbody>
        <?php

            $records = $conn->prepare('SELECT game_name, duration, played_date FROM played
WHERE player_id = :player_id');
            $records->bindParam(':player_id', $_SESSION['user_id']);
            $records->execute();
            $results = $records->fetchAll();

            foreach($results as $result)
            {
                echo "<tr>";
                echo "<td>" . $result['game_name'] . "</td>" . "<br>";
                echo "<td>" . $result['duration'] . "</td>" . "<br>";
                echo "<td>" . $result['played_date'] . "</td>" . "<br>";
                echo "</tr>";
            }
        ?>
    </tbody>
</table>

```

## 4. Functional Components

### 4.1 Use Cases/Scenarios

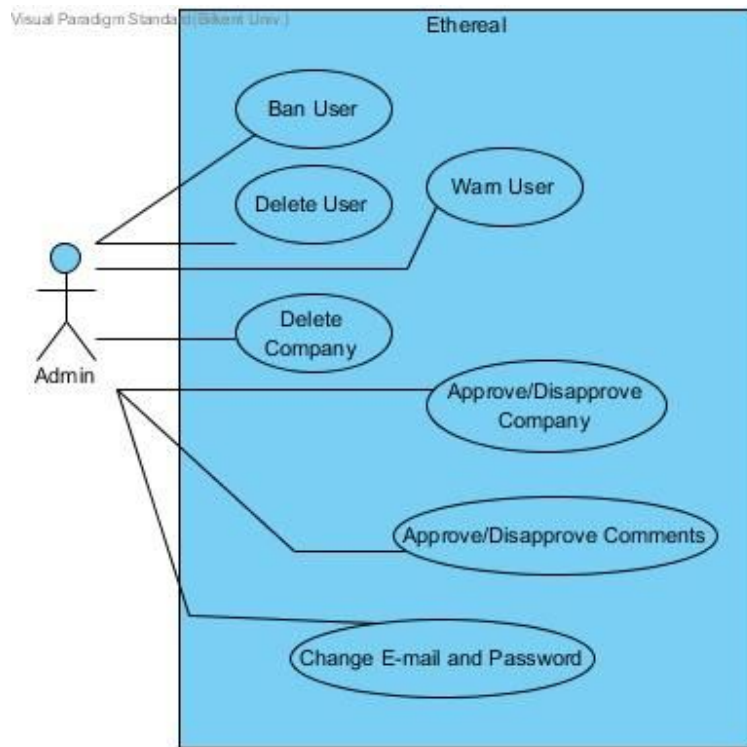
Ethereal has three different types of users. Use cases are as follows.

#### 4.1.1 Admin

The admins are the administrators of Ethereum. This means that an admin should be able to control users, their activities and update content of the website if necessary.

##### **Admin**

- An admin can ban a user
- An admin can delete a user
- An admin can warn a user
- An admin can delete company
- An admin can approve/disapprove following content:
  - An admin can approve/disapprove the company
  - An admin can approve/disapprove the comments
- An admin can change email and password



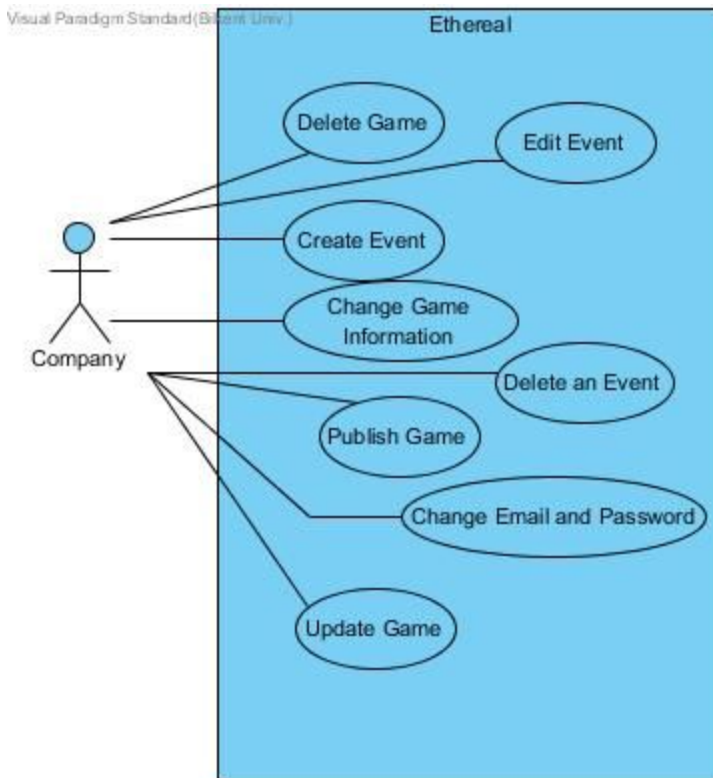
#### 4.1.2 Company

A company is the type of account which uploads games to the system, creates events for the games and edits their information. They gain profit from their game and they can change the game information as they like.

##### **Company**

- A company can publish a game.
- A company can delete a game.
- A company can change information about the game.
- A company can update their games.
- A company can create an event
- A company can edit the existing event
- A company can delete an event

- A company can change email and password



#### 4.1.3 Player and Premium Player

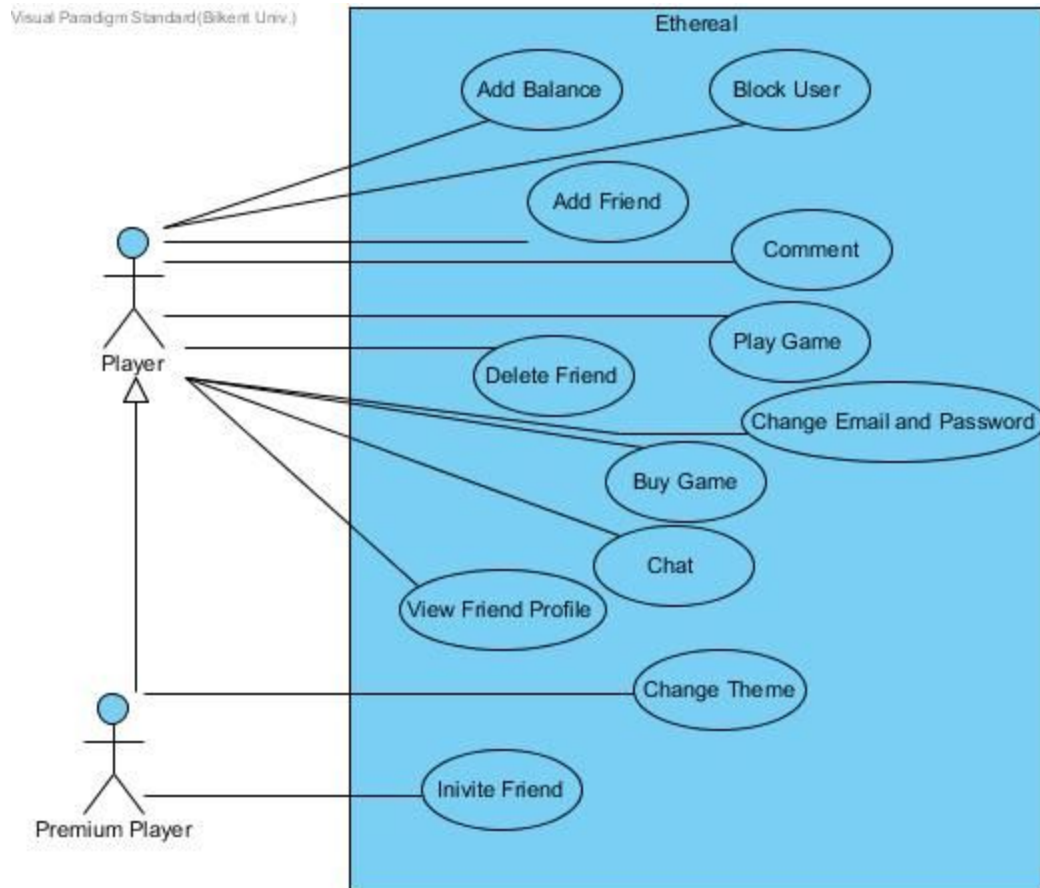
Players are the main actors of Ethereal. They can play games, buy games, chat and add their friends to the system. Also, there's a premium account type is available which adds new features to a player as being able to change themes and ability to invite friends to a game.

##### **Player**

- A player can add balance.
- A player can block/unblock another player.
- A player can add/unfriend a friend.
- A player can comment on games.
- A player chat with a friend.
- A player can view his/her friend's profile.
- A player can buy a game.
- A player can play a game.
- A player can update their profile into the premium player.
- A player can change his/her email and password.

### Premium Player

- A premium player can do everything that a player can do.
- A premium player can change the theme of the system.
- A premium player can invite his/her friends to a game to play together.



## 4.2 Algorithms

Different algorithms will be used for different types of users.

### 4.2.1 Player Related Algorithms

We want to keep all players' information related to the games and game categories they play. First, players will be able to see all games. By the time, the games will get categorized by corresponding game category. Later, players will be able to see their and their friends' most played games and categories. In addition, premium players will be able to see the statistics of their invited games (i.e to which game they have been invited the most). By using the provided information, players will be able to suggest games to others or invite other players into the games that they play the most.



#### **4.2.2 Game Related Algorithm**

Since game is the main product of Ethereum, it will be kept stable and secure. The data-management system will keep track of game statistics such as the most sold/played games.

#### **4.2.3 Logical Requirements**

To prevent and minimize the logical errors some precautions will be taken.

- The date attributes of our system will be restricted to prevent possible logical mistakes. For instance, when a company publish a game or when the user share a comment, published\_date and comment\_date attributes of the Game and Comment tables will be filled automatically with the date of that day. In addition, when Company will create an event, chosen start\_date and end\_date can not be a past date. Additionally, end\_date cannot be chosen earlier than start\_date of the event. Therefore, start\_date and end\_date are boundary points in Event table and company should choose a date according to these restrictions.
- The values such as price, age\_limit and balance should not be negative. By taking this precaution, the system will prevent possible consequences such as a player buys a game even if he/she does not have enough money (balance becomes negative), a player enters a game with larger age\_limit restriction than his/her own age (age - age\_limit should more than zero in order to enter the game). Additionally, birth\_date of a player should be realistic (birth\_date cannot be more than less than 1918, for example).

#### **4.3 Data Structures**

We are using built-in data types of SQL such as int, varchar, char, float, boolean, date etc. There will not be any additional data structure used.

## 5.User Manual

### 5.1 Index Page

**Process:** From Index page user can choose its type. By clicking the relevant button user can continue as admin, company or player .

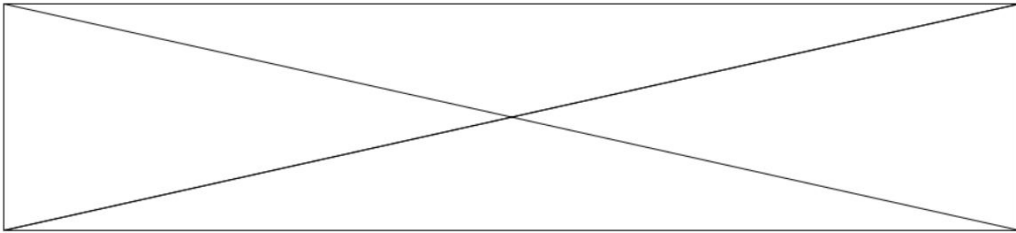
WELCOME TO ETHEREAL



### 5.2 Admin Login Page

**Process:** Admins can log in with their passwords and usernames. To the admins who forgot their password a forgot password option is available.

## ADMIN LOGIN



## WELCOME TO LOGIN PAGE

Username:

Password:

Login

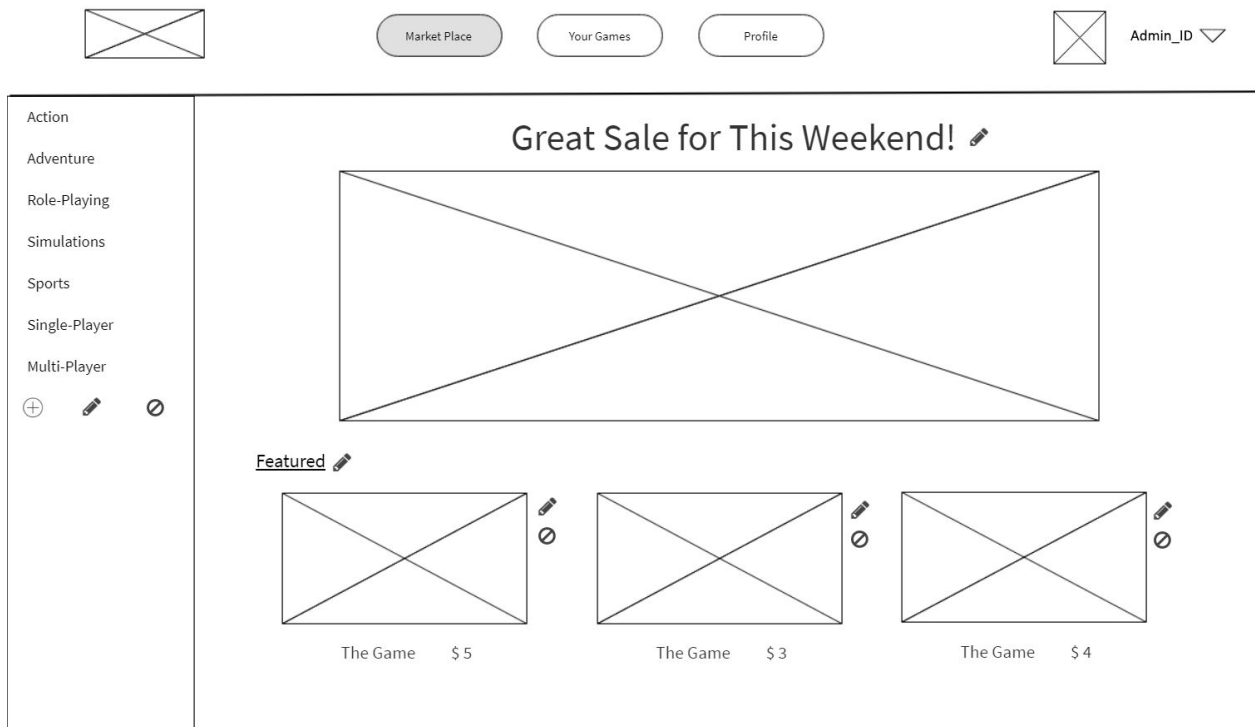
[Forgot password?](#)

[Company Login](#)

[Player Login](#)


### 5.3 Admin Marketplace

**Process:** Admin can see the event list and category list from their marketplace page. By clicking the manage companies and manage events button admin can go the manage companies and events page.



## 5.4 Admin Profile


**Process:** Admin can see their information in their profile page.



Market Place

Games

Profile

 Admin\_ID ▾

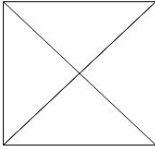
---

Manage Companies

Manage Users

Manage Comments

Account Settings



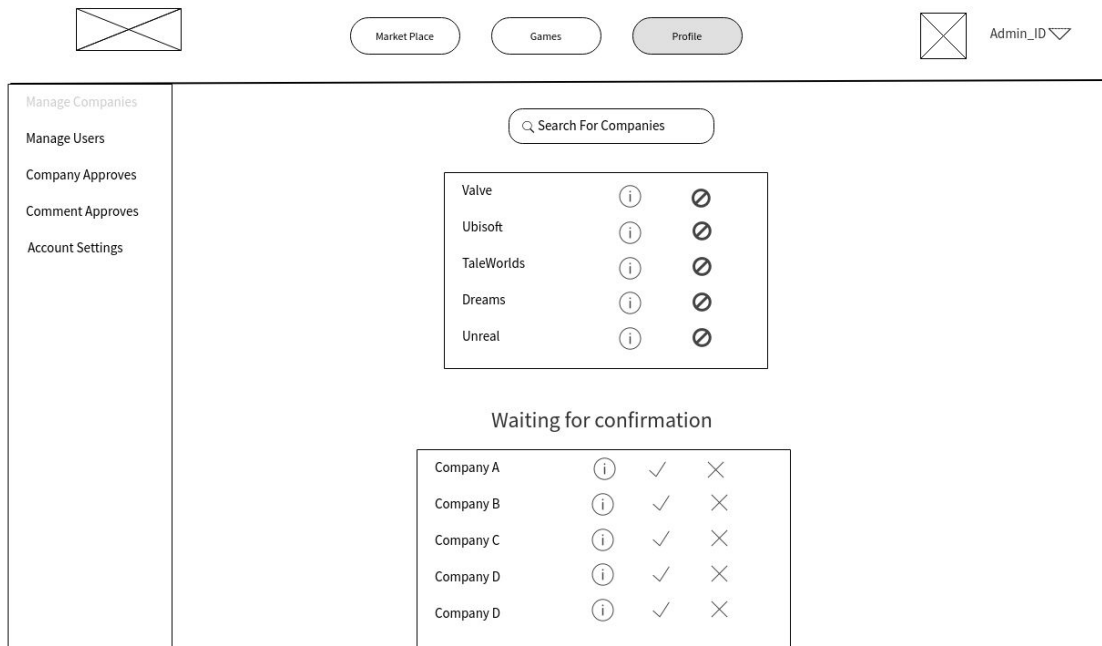
Admin\_ID

Email Address:

Password:

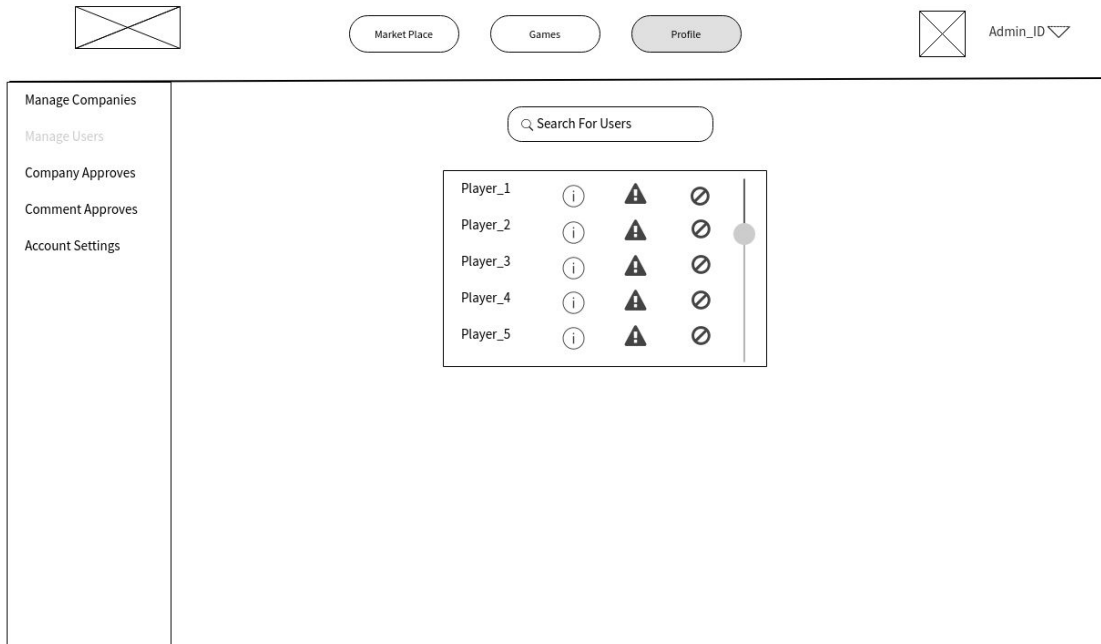
## 5.5 Admin Manage Companies

**Process:** From admin manage companies page admins can confirm the new companies and ban the existing companies.



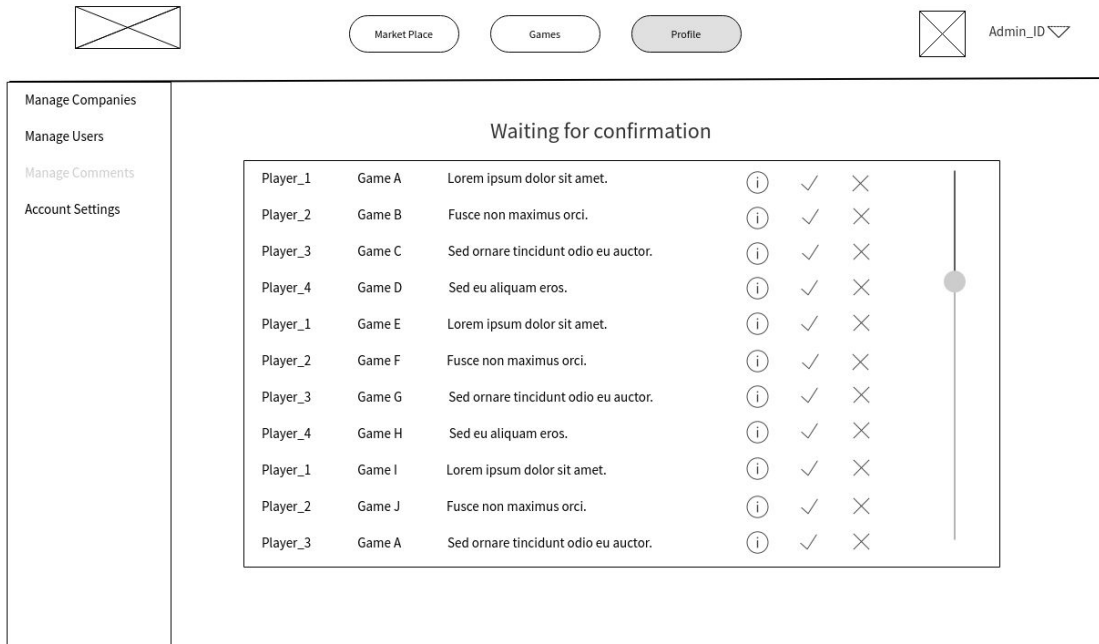
## 5.6 Admin Manage Players

**Process:** From admin manage users page admins can warn and ban the players. The player which is warned 3 times will be banned by admin.



## 5.7 Admin Manage Comments


**Process:** From admin manage comments, admins can see the comments made by players to the corresponding games. Admins can confirm the new comments and delete the comments which has unethical elements from this page.




## 5.8 Admin Manage Categories

**Process:** From admin manage categories, admins can see the categories added by companies to the corresponding games. Admins can change the categories and add new categories from this page.





Market PlaceGamesProfile

Admin\_ID ▾

---

Manage Companies  
Manage Users  
Manage Comments  
Manage Categories  
Account Settings


### Manage Categories

Add new category :


Change category :

## 5.9 Admin Account Settings

**Process:** When admin opens account settings page they can change their email and password.

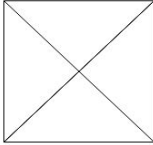


Market PlaceGamesProfile

Admin\_ID ▾

---

Manage Companies  
Manage Users  
Manage Comments  
Account Settings



Admin\_ID

Email Address:

Old Password:

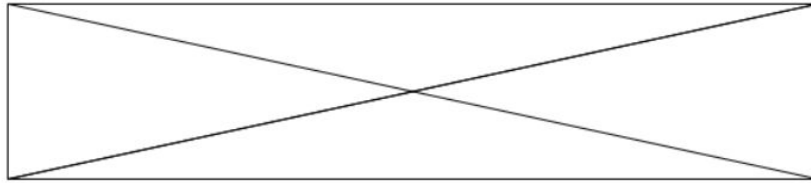
New Password:

New Password Again:

Save Changes

## 5.10 Company Sign Up Page

**Process:** The company should fill all of the informations in the sign up page. The new Customer added to database. The status of the company will be 0 in the Approves table until admin approve the company. After the approval company can be log in to the website and upload its games.



## COMPANY SIGN UP PAGE

Username:

Password:

E-mail:

Company Name:

Web page:

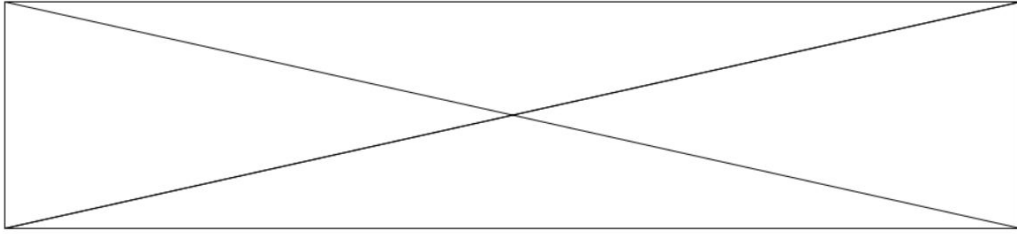
Address:

Description:

### 5.11 Company Login Page

**Process:** Companies can log in with their passwords and usernames. To the companies who forgot their password a forgot password option is available.

## COMPANY LOGIN



## WELCOME TO LOGIN PAGE

Username:

Password:

Login

[Forgot password?](#)

[Player Login](#)

[Admin Login](#)

## 5.12 Company Marketplace

**Process:** Companies can see their games from its marketplace. By clicking the add event and manage event buttons company can add and manage their events.

## 5.13 Company Event Addition

**Process:** Companies can add event from this page. All information of event should be filled in and company should choose the games that the event will be applied to.

The screenshot shows a web interface for adding an event. At the top, there is a navigation bar with a logo on the left, three buttons labeled 'Market Place', 'Games', and 'Profile' in the center, and a user profile icon and 'Admin\_ID' dropdown on the right. On the left side of the main content area, there is a vertical sidebar with two links: 'Add Event' (highlighted) and 'Manage Events'. The main content area is titled 'Add Event' and contains the following form fields:

- Event Name:** A text input field.
- Description:** A text input field.
- End Date:** A date input field with the value '01/2019'.
- Discount Percent:** A text input field with the value '10'.
- Add Game to Event:** A section with two columns of game selection options.
  - Left column: Game A, Game B, Game C, and Game D, each with a '+' button.
  - Right column: Game G and Game E, each with a '-' button.
- Add Event:** A button at the bottom center of the form.

## 5.14 Company Event Management

**Process:** Companies can delete its events from this page. Also companies can manage events by changing their names, dates and descriptions

Market Place

Games

Profile

Admin\_ID ▾

Add Event

Manage Events

Manage Events

Event_1		
Event_2		
Event_3		

Event Name:

Description:

End Date:

01/2019

Discount Percent:

10

Add Game to Event:

Game A	
Game B	
Game C	

Game G	
Game E	


Update

### 5.15 Company Games


**Process:** Companies can add, manage and update games by clicking the related buttons. Company can also see their games' statistics by clicking the game statistic button.

### 5.16 Company Game Addition

**Process:** Companies can add game through add game page by filling the necessary informations.



Market PlaceGamesProfile

Company\_ID ▾

Add Game  
Manage Game  
Update Game

### Add a New Game


Game Name:Game A

Price:\$5

Developer:SkyLabs

Description:

Source Code:



Add a Picture

Category:

Select ▾

Add a File

Publish Game

## 5.17 Company Update Game

**Process:** Companies can update their games from this page by inserting new information and the new source code of the game.

Market Place

Games

Profile

Company\_ID ▾

Add Game

Manage Game

Update Game

Game Updates

Pick a Game: 

Game A ▾

Change Log:

New Source Code:

Add a File

Publish New Update

## 5.18 Company Manages Game

**Process:** Companies can manage game through changing informations of the game.



Market Place

Games

Profile

Company\_ID ▾

Add Game

Manage Game

Update Game

Manage Game

Game Name: 

Game A ▾

Price: 

55

Developer: 

SkyLabs

Description:

Category: 

Select ▾

Save Changes

Delete Selected Game

Change a Picture

## 5.19 Company Game Statistics

**Process:** Companies can see the game statistics about their games. Companies can see the Number of Players Playing its games, Number of Players Bought its Games, the Female Players and Male Players played the game and the country of the players. In addition, companies can see for each of their game how many player plays from which age.

## Company Profile

### Market Games Profile

Add Game  
Manage Game  
Update Game  
Game Statistics

Number of Players playing your games: 10  
Number of Player bought your games: 23  
Number of Female Players: 3  
Number of Male Players: 7

Countries
Lebanon
Kenya
Vatikan
Zimbabwe

Age	Game	Player#
18	G1	4
14	G2	5

#### 5.20 Company Profile Information

**Process:** Companies can see and change their information from their profile page. Profile Page includes company webpage, zip, description, state, city, distinct, picture and name of the company.

Market Place

Games

Profile

Company\_ID ▾

Information

Account Settings

Company\_ID

Company Name

Web Page: company.com

Zip: 06800

Distinct: Cankaya

City: Ankara

State: Turkey


Description:

Text Here....


Save

## 5.21 Company Account Settings

**Process:** When companies open account settings page they can change their email, passwords. The database will not be modified until player clicks the save changes button.

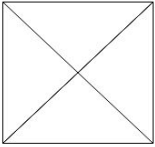


Market PlaceGamesProfile

Company\_ID ▾

---

Information  
Account Settings



Company\_ID  
Company Name

Email Address:

Old Password:

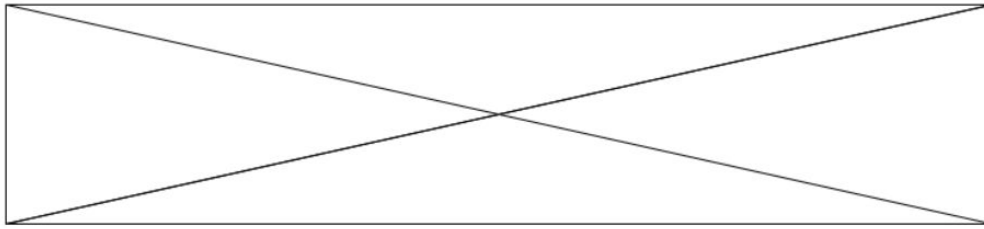
New Password:

New Password Again:

Save Changes

## 5.22 Player Sign Up Page

**Process:** The player should fill all of the informations in the sign up page. The new Player is added to database.



## USER SIGN UP PAGE

Username:

Password:

E-mail:

Full Name:

Birth Date:

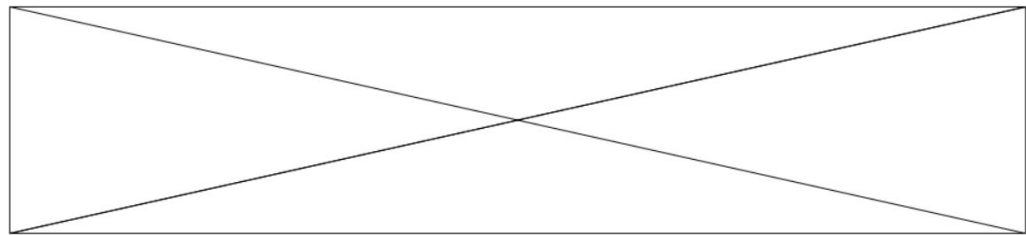
Gender:

Country:

Biography:

## 5.23 Player Login Page

**Process:** Players log in with their passwords and usernames. To the players who forgot their password a forgot password option is available.



### WELCOME TO LOGIN PAGE

Username:

Password:

Login

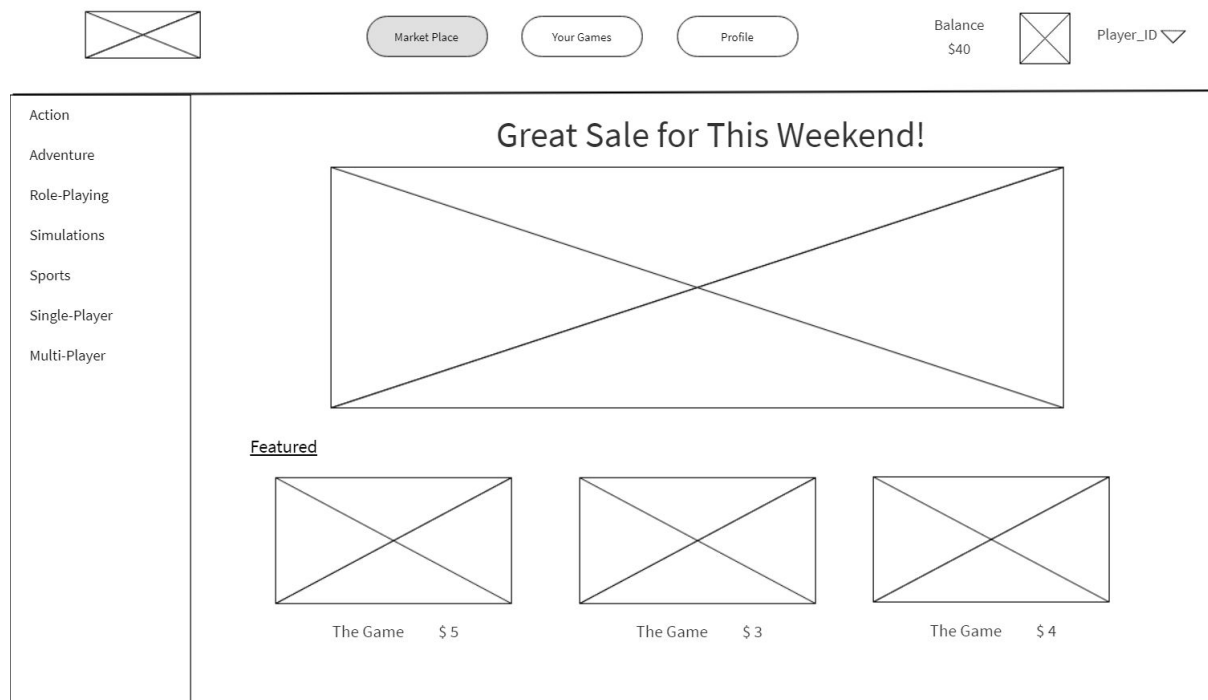
[Forgot password?](#)

[Company Login](#)

[Admin Login](#)

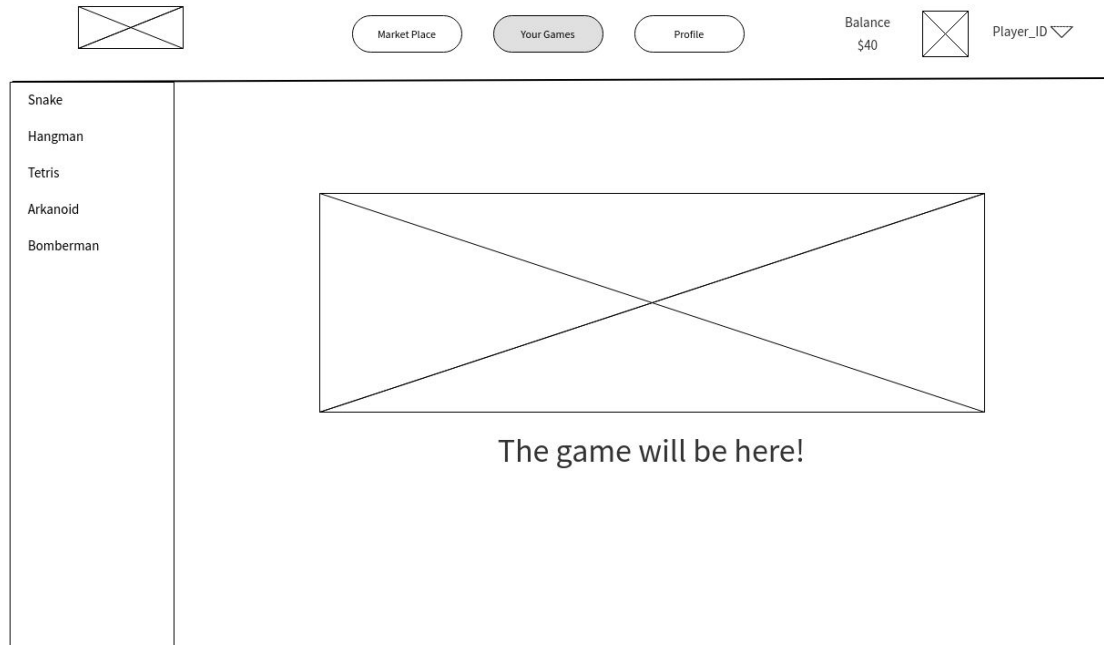
## 5.24 Player Marketplace

**Process:** After players log in their accounts they can see the discount events of the game and see their balances, buy a new game in user marketplace page. In addition this pages also provides players a link to the list of the games in each category, a link to list of the games they bought and a link to go to their profiles. Players can also see their balance from the top of page.



## 5.25 Player Games


**Process:** By using “Games” button, players can reach the games they bought.



## 5.26 Player Profile

**Process:** Players can see their information from their profile page. Profile Page includes birth date, country, gender and biography of the player. In addition, player can see its friends from the right of the page. By clicking the information buttons next to the friends name players can go to the profiles of their friends.






Market Place

Your Games

Profile

Balance  
\$40



Player\_ID ▾

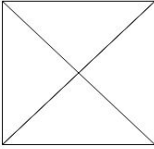
---

Information

Game History

Add Funds to Wallet

Account Settings



Player\_ID

Full Name

Birth Date: 01/01/1999

Country: Turkey

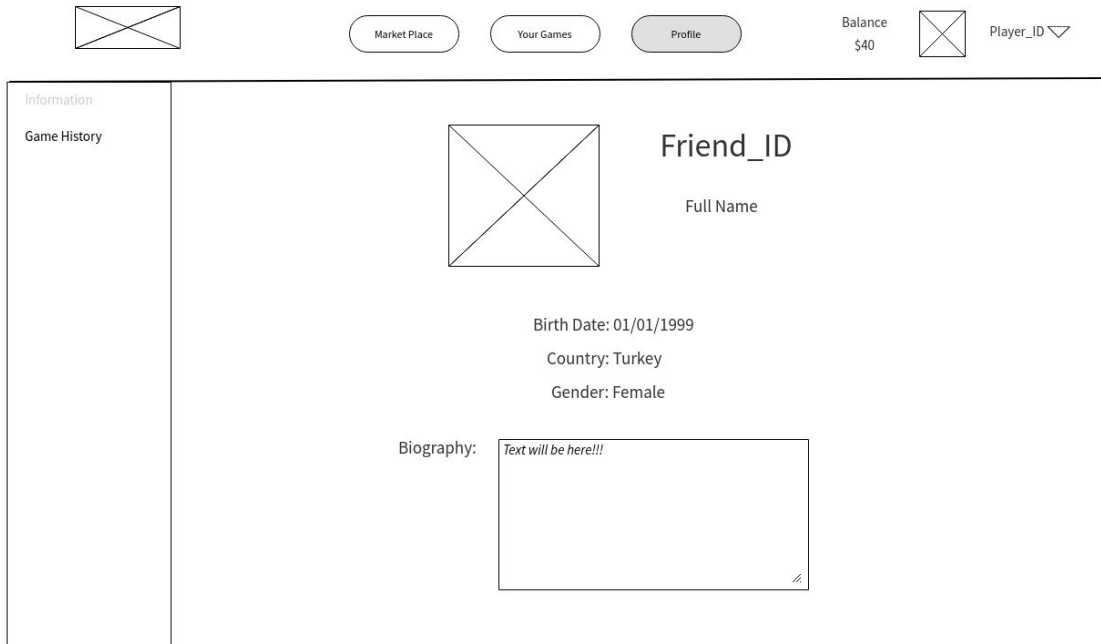
Gender: Female

Biography: I came from Lebanon. I love playing games ;))

Save

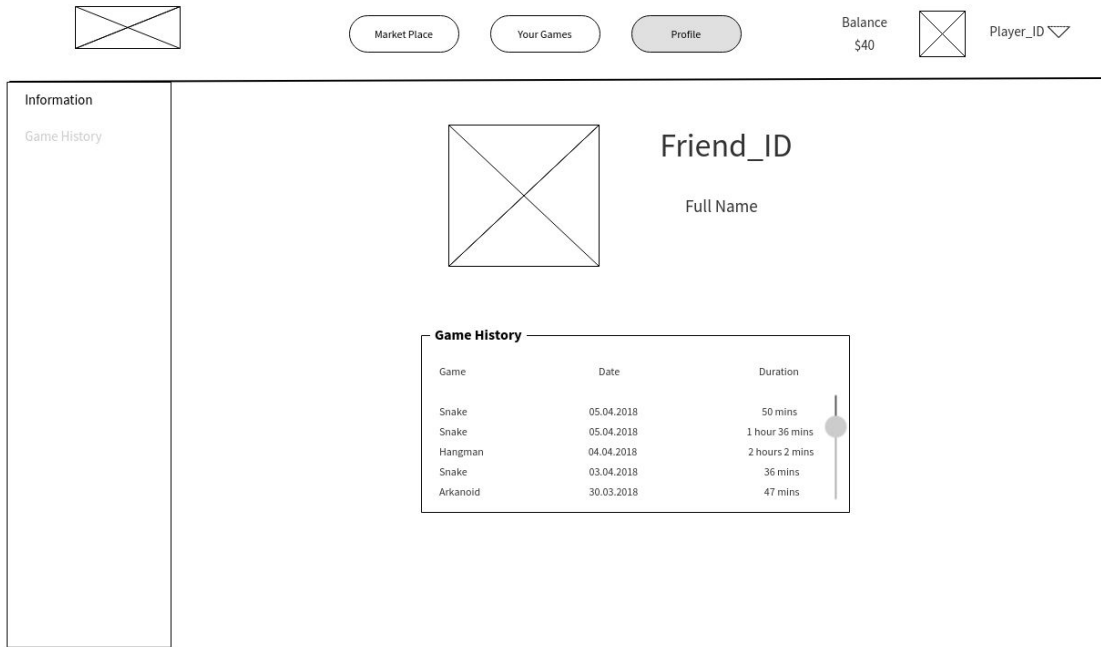
## 5.27 Friend Profile Information

**Process:** Players can see their friends' information by looking their profile. Friend Profile Page includes birth date, country, gender and biography of the friend.



## 5.28 Friend Profile History

**Process:** Each player can look at their friends' game histories. Game history page includes the latest played games, the date and duration of the play.



## 5.29 Player Profile Information

**Process:** Players can see and change their information from their profile page. From this page players can change their birth date, country, gender and biography of the player.

Market Place

Your Games

Profile

Balance  
\$40

Player\_ID ▾

Information  
Game History  
Add Funds to Wallet  
Account Settings

Player\_ID

Full Name

Birth Date: 01/01/1999

Country: Turkey

Gender: Female

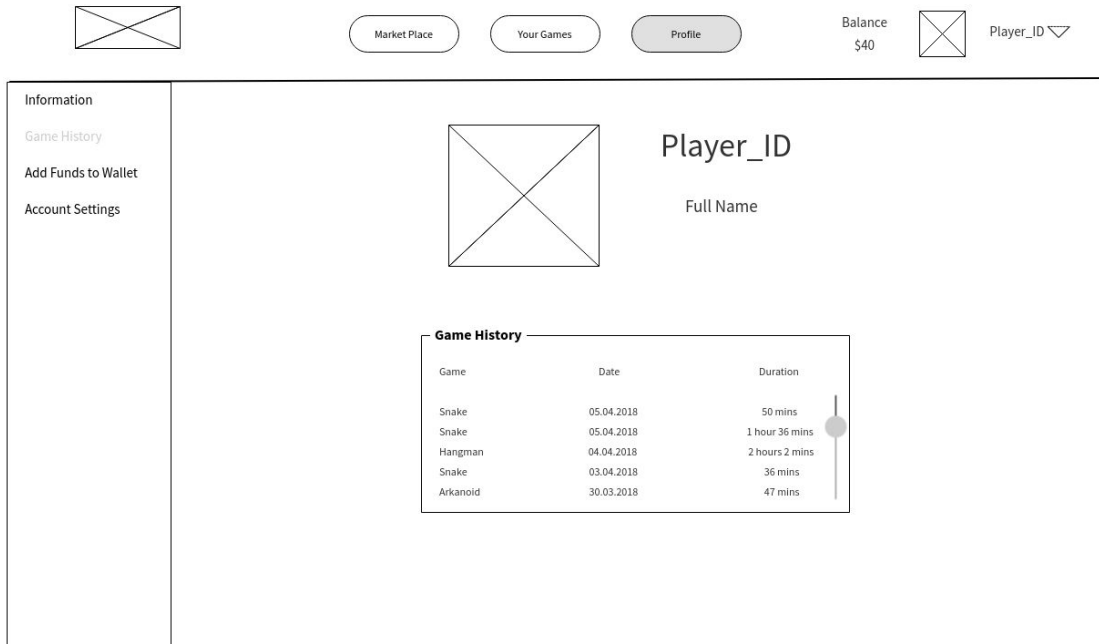
Biography: 

Text will be here!!!

Save


### 5. 30 Player Game History

**Process:** Each player can look at their game history. Game history page includes the latest played games, the date and duration of the play. In addition, from the table at the left players can go to their information page, funds page and account setting page.




### 5.31 Add Funds to Wallet

**Process:** Player can increase their balance by filling the credit card information in this page.

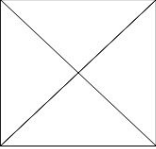


Market PlaceYour GamesProfile

Balance  
\$40

 Player\_ID ▾

Information  
Game History  
Add Funds to Wallet  
Account Settings



Player\_ID  
Full Name

☒ Add funds☐ Premium

Payment

Card Number:

Card Holder:

Expiration Date:


Amount:

Security Number:

Add Fund

### 5.32 Player Account Settings

**Process:** When players open account settings page they can change their email, passwords. They can also see their status(premium or normal) and number of warnings that is done by admin. The database will not be modified until player click the save changes button.




Market Place

Your Games

Profile

Balance  
\$40

 Player\_ID ▾

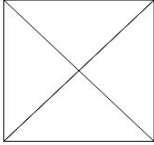
---

Information

Game History

Add Funds to Wallet

Account Settings



Player\_ID

Full Name

\*Premium Status\*

You have 0 warns by admins.

Email Address:

Old Password:

New Password:

New Password Again:

Save Changes

### 5.33 Chat pop-up

**Process:** Players can see their friends in friends pop-up page. In addition from this page they can see the status of their friends and message their friends through the pop-up screen in this page. Additionally, players can block,delete and show the profile of their friends by using this page.



Market Place

Your Games

Profile

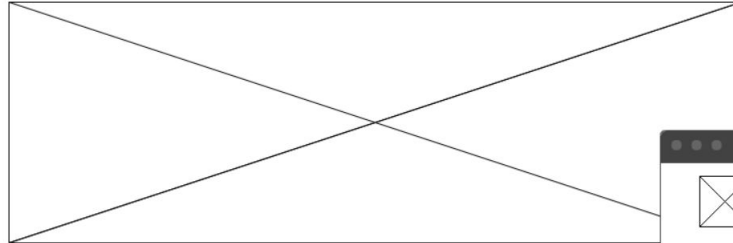
Balance  
\$40



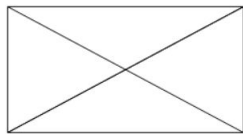
Player\_ID ▾

Action  
Adventure  
Role-Playing  
Simulations  
Sports  
Single-Player  
Multi-Player

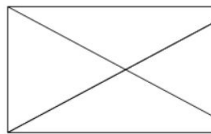
## Great Sale for This Weekend!



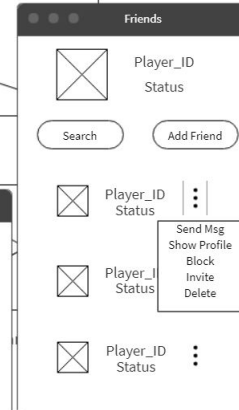
### Featured



The Game \$ 5



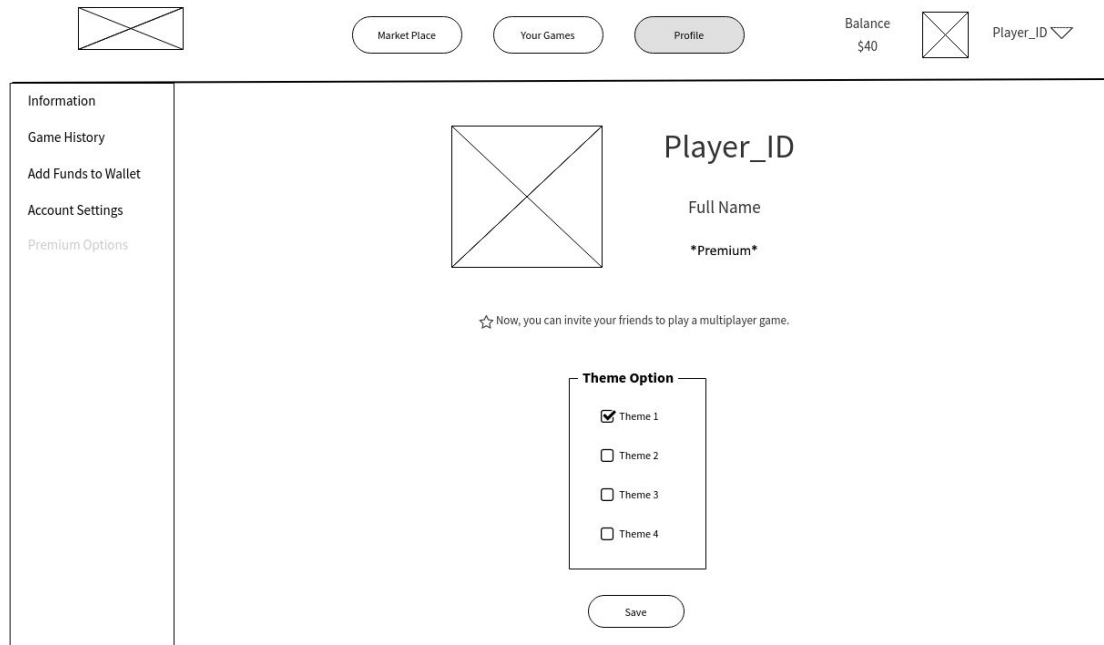
The Game \$ 3






## 5.34 Premium Player Profile

**Process:** Premium players can update their background themes



The image shows a user interface for a premium player's profile. At the top, there is a navigation bar with a placeholder icon, three buttons labeled 'Market Place', 'Your Games', and 'Profile', a balance indicator showing '\$40', and a dropdown menu for 'Player\_ID'. On the left side, there is a vertical sidebar menu with links: 'Information', 'Game History', 'Add Funds to Wallet', 'Account Settings', and 'Premium Options' (which is highlighted). The main content area displays the player's profile information, including a placeholder for a profile picture, the text 'Player\_ID', 'Full Name', and a premium status indicator '\*Premium\*'. Below this, a message states '☆ Now, you can invite your friends to play a multiplayer game.' Underneath the message is a 'Theme Option' section with a list of four themes: 'Theme 1' (selected with a checked checkbox), 'Theme 2', 'Theme 3', and 'Theme 4'. A 'Save' button is located at the bottom of the theme selection area.

Market PlaceYour GamesProfileBalance  
\$40Player\_ID ▾

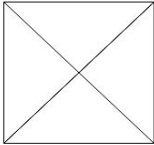
Information

Game History

Add Funds to Wallet

Account Settings

Premium Options



Player\_ID

Full Name

\*Premium\*

☆ Now, you can invite your friends to play a multiplayer game.

Theme Option

☒ Theme 1

☐ Theme 2


☐ Theme 3

☐ Theme 4

Save

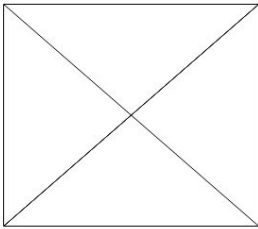
## 5.35 Player Writes Comment

**Process:** Players can add a comment under a game and wait for it to be approved. They can also see the comment list for a game.


Market Place
Your Games
Profile
Balance  
\$40
Player\_ID ▾

Action  
Adventure  
Role-Playing  
Simulations  
Sports  
Single-Player  
Multi-Player

## The Game



Action Game +13

Bought Already

Price: 4\$ 5\$

Easter Event Discount %20  
until : 04/04/2018

**Information**

Developer: SkyLab  
Publisher: Company  
Published Date: 01/01/2018

Last Update: 23/03/2018

Update Change Log:  
Change log of the game will be here. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer mollis lorem nec diam tempor sodales.

**Comments**

Player_1	It is pretty good game!	👍 10 👎 1
Player_2	So nice game!	👍 8 👎 3
Player_3	I m totally in this game!	👍 6 👎 2
Player_4	I didnt like. It is so buggy.	👍 1 👎 11

Make a comment:

Text here....

Submit

## 6. Advanced Database Components

### 6.1 Views

### 6.2 Constraints

- Players cannot login the website if they are banned.
- The company which is disapproved by admin is inhibited to login the system.
- The company in the disapproved list cannot sign up with same company name once more.
- Admin cannot ban a banned player once again.
- The price of the game in the event list cannot be larger than its original price.
- Every game should have a company and category.
- Every information has a corresponding player.
- Users (admins, companies, players) cannot see others passwords.
- The descriptions and game lists can be seen by all people, however, to play a game user should buy the game.
- Companies cannot add an additional category for their game, if a company creates a game with a category which is not found in the category choices, that company should contact the admin, if the admin approves the new category, it will be added to the category list.

- Players cannot message or play with a person that she/he is blocked by.
- A player cannot buy a game which has a price larger than Player's balance.
- A player cannot like/dislike comments more than one time.
- A player cannot buy a game which exceeds her/his age limit.
- Game's published date and message date cannot be postdated.

### **6.3 Triggers**

- When a game is updated, update date should be changed.
- When Likes relation is updated, like\_count or dislike\_count should be updated.
- When a company creates an event for their games the discount percentages should correspond to the games' unit price.
- When a player adds a fund to the balance, the balance should increase by fund amount.
- When a player buys a game the balance of the player should decrease by the game price amount.

### **6.4 Stored Procedures**

- Players will be notified when they get an invitation.
- Players will be notified when any of admins warns them.
- Players will be notified when another player adds them.
- Players will be notified when they receive a message.
- Admins will be notified when a company wants to sign up to the marketplace.
- Admins will be notified when a new comment is made under any game.
- Admins will be notified when a new event is added to the marketplace.

## **7. Website**

This design report and other activities of the project are available here:

<https://github.com/mertosmandy/Ethereal>