



Università  
Ca' Foscari  
Venezia

# Nopaynogame

Progetto di Basi di Dati Modulo 2

Anno Accademico  
2017/1018

## **Gruppo**

*Elia Ferrotti, 847059*

*Fabrizio Ortotti, 848856*

*Alessia Michela di Campi, 861844*

*Andrea Corazza, 843173*

# 1 Indice dei contenuti

1.	Introduzione.....	3
2	Descrizione del Progetto.....	4
2.1	Scelta del Progetto.....	4
2.2	Modifica e/o aggiunte alla consegna .....	5
3	Database .....	6
3.1	Schema Concettuale.....	6
3.2	Schema Relazionale .....	7
3.3	Creazione del Database .....	8
3.4	Popolamento del Database .....	10
4	Diagramma Applicazione Web.....	11
5	Interfacce .....	12
5.1	Header .....	12
5.2	Home .....	12
5.2.1	Genere .....	13
5.2.2	Console .....	13
5.2.3	Catalogo Completo.....	14
5.2.4	Ricerca .....	14
5.3	Gioco.....	15
5.3.1	Card Gioco.....	16
5.3.2	Descrizione .....	17
5.3.3	Recensioni .....	17
5.3.4	Giochi Consigliati .....	18
5.4	Log In .....	19
5.4.1	Accesso .....	19
5.4.2	Registrazione .....	19
5.5	Profilo.....	20
5.5.1	Informazioni.....	20
5.5.2	Ordini Effettuati .....	21
5.5.3	Review E Voti .....	23
5.5.4	Il Mio Conto .....	24
5.6	Carrello .....	24
5.6.1	Step 1 – Riepilogo Carrello.....	24
5.6.2	Step 2 - Spedizione.....	24
5.6.3	Step 3 – Conferma Acquisto .....	25
5.6.4	Step 4 – Conferma Ricezione Ordine .....	25
5.7	Amministrazione .....	26
5.7.1	Magazzino.....	26
5.7.2	Aggiunta Nuovo Gioco .....	27
5.7.3	Gestione Ordini.....	28
5.7.4	Gestione Profili .....	29

5.7.5	Aggiunta Nuovo Profilo .....	30
6	Struttura del Progetto.....	31

# 1. Introduzione

Il presente documento vuole documentare e presentare il progetto del sito NoPayNoGame svolto dal gruppo presentato in copertina.

Il documento presenterà una descrizione della scelta di progetto con eventuali modifiche inserite, la presentazione del modello dei dati dall'analisi concettuale all'analisi relazionale alla realizzazione, la struttura e la descrizione dell'applicazione web, l'analisi delle query effettuate per gestire le diverse funzionalità.

## 2 Descrizione del Progetto

Lo scopo del progetto è la creazione di un negozio elettronico di videogiochi che offre diverse funzionalità in base al tipo di utilizzatori. I possibili utilizzatori previsti sono: consumatori, magazzinieri, amministratori e superutenti.

Nel caso l'utilizzatore sia un Utente/Consumatore non registrato, le feature offerte sono:

- La visualizzazione del catalogo di giochi;
- Le ricerche sul catalogo stesso;
- La visualizzazione del singolo gioco;
- La creazione un nuovo profilo;
- L'accesso alla piattaforma.

Nel caso l'utilizzatore che ha effettuato l'accesso sia un Consumatore normale, le feature offerte sono:

- L'accesso ai cataloghi e le ricerche;
- La visualizzazione del singolo gioco;
- La creazione di una recensione del gioco se già acquistato;
- L'aggiunta al carrello degli articoli disponibili;
- La visualizzazione del carrello;
- La modifica/cancellazione del carrello;
- La creazione di un ordine;
- La modifica/cancellazione di un ordine;
- La visualizzazione della pagina del suo profilo;
- La modifica dei dati del proprio profilo;
- La visualizzazione di tutte le recensioni scritte;
- La modifica/cancellazione delle recensioni scritte.

Nel caso l'utilizzatore che ha effettuato l'accesso sia un Magazziniere, le feature offerte sono:

- L'accesso alla pagina di amministrazione;
- La modifica/cancellazione dei giochi correnti;
- L'aggiunta di nuovi giochi.

Nel caso l'utilizzatore che ha effettuato l'accesso sia un Amministratore, le feature offerte sono quelle del magazziniere con l'aggiunta di:

- Gestione stato ordini;
- Cancellazioni ordini.

Nel caso l'utilizzatore che ha effettuato l'accesso sia un Superutente, le feature offerte sono quelle dell'amministratore con l'aggiunta di:

- Modifica dei profili;
- Creazione di nuovi profili.

Il progetto è costituito da due parti: un database per la gestione di tutti i dati, utenti, giochi e un sito web. Per comodità organizzative è stato deciso di utilizzare Altervista.org come host per la parte web e di database. Il progetto è consultabile al seguente link:

<http://nopaynogame.altervista.org/nopaynogame/>.

### 2.1 Scelta del Progetto

La traccia scelta per lo sviluppo del progetto è la seguente:

#### **NEGOZIO ELETTRONICO DI VIDEOGIOCHI (3-4)**

Base di dati:

**Videogiochi** (titolo, console, prezzo, quantità disponibile in magazzino). **Utenti** con nome e cognome, indirizzo, telefono, login e password. **Ordini**: utente, titoli ordinati, data, tipo di pagamento.

#### **Accesso senza login:**

Catalogo dei videogiochi, visualizzazione e ricerca. Registrazione di un utente.

#### **Accesso con login:**

Ordine.

Cancellazione di un'ordinazione.

**Accesso con login di amministratore:**

Inserimento e modifica dei dati, degli utenti e degli ordini.

## 2.2 Modifica e/o aggiunte alla consegna

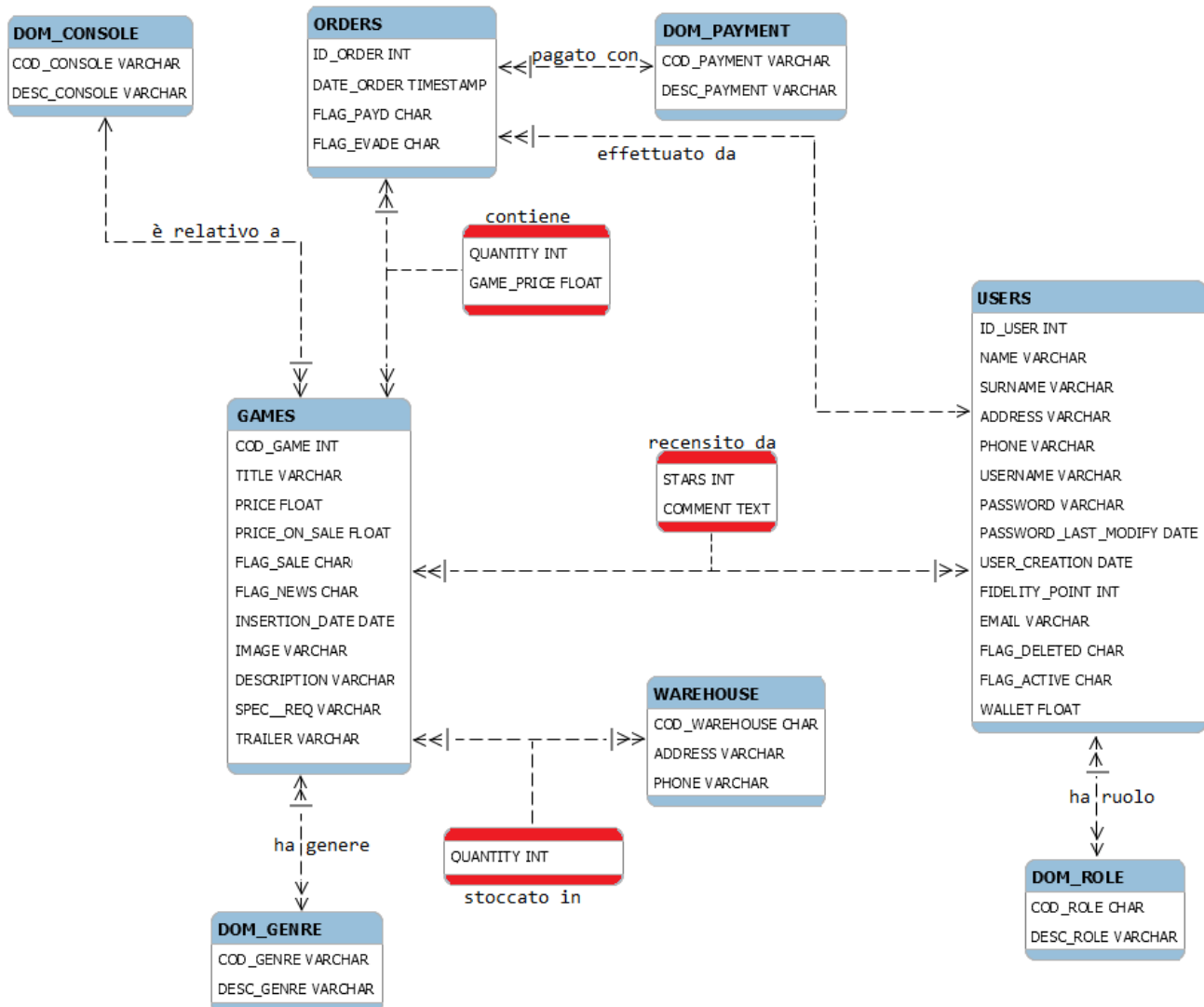
Abbiamo scelto di modificare la traccia del progetto ampliandone le funzionalità offerte come segue:

- È presente la gestione dei magazzini: è prevista una tabella magazzino che tiene traccia dei dati del magazzino stesso e delle quantità di ogni singolo gioco in esso presente. In questo modo è possibile gestire la presenza di molteplici magazzini;
- È presente un sistema di recensioni ai prodotti costituite da un voto (1-5 stelle) e un commento opzionale;
- È presente la gestione dell'elenco dei generi riferiti a ciascun gioco;
- È presente un campo portafoglio ed un campo punti fedeltà per gestire l'accredito verso l'utente in caso di eliminazione dell'ordine e per offrire al cliente un sistema di bonus acquistabili con i punti (ricaricati attraverso gli acquisti) che permettono di ricaricare il portafoglio online;
- È previsto un sistema di gestione dello stato dell'ordine (pagato - evaso):
  - Utente: quando un ordine risulta ancora non pagato, l'utente ha la possibilità di eliminarlo e/o modificarlo in base alle disponibilità dei giochi nel magazzino;
  - Amministratore: ha la possibilità di registrare il pagamento e solo a quel punto di poter evadere l'ordine. Nel caso in cui l'ordine sia non-pagato e/o non-evaso, l'amministratore ha la possibilità di eliminare l'ordine ed effettuare un eventuale rimborso;
- Sono presenti più ruoli rispetto a quello proposto: Utente normale, Magazziniere, Amministratore, Superuser. Ogni ruolo ha delle feature diverse, come spiegato nella [Descrizione del Progetto](#);
- È presente una gestione degli utenti su più passi: un utente può avere un account attivo/non attivo e cancellato/non cancellato.

## 3 Database

### 3.1 Schema Concettuale

Di seguito lo schema concettuale del database:



Da come è possibile vedere dall'immagine, abbiamo scelto di completare la consegna fornitaci con la creazione di una base di dati costituita prevalentemente da quattro grandi tabelle:

- Games: tabella per la gestione dei vari giochi;
- Users: tabella per la gestione degli utenti;
- Orders: tabella per la gestione degli ordini;
- Warehouse: tabella per la gestione dei magazzini.

A questo insieme di tabelle sono state aggiunte poi delle tabelle "DOM\_nome" che fungono da domini per determinati valori poiché una volta create e popolate, non vengono più toccate con inserimenti da parte degli utenti.

Le relazioni sono rappresentate da delle frecce che dimostrano la tipologia di relazione (1:n, n:m) e la sua totalità o meno (| ). I riquadri rossi indicano delle relazioni che necessitano di attributi.

Per ogni gioco interessa sapere: il titolo, il prezzo, il prezzo in saldo, quando è in saldo, quando è una novità, la data di inserimento, l'immagine, la descrizione, la specifica dei requisiti, il trailer, l'elenco dei generi, la piattaforma, la quantità presente in ciascun magazzino.

Per ogni magazzino interessa sapere: l'indirizzo e il numero di telefono.

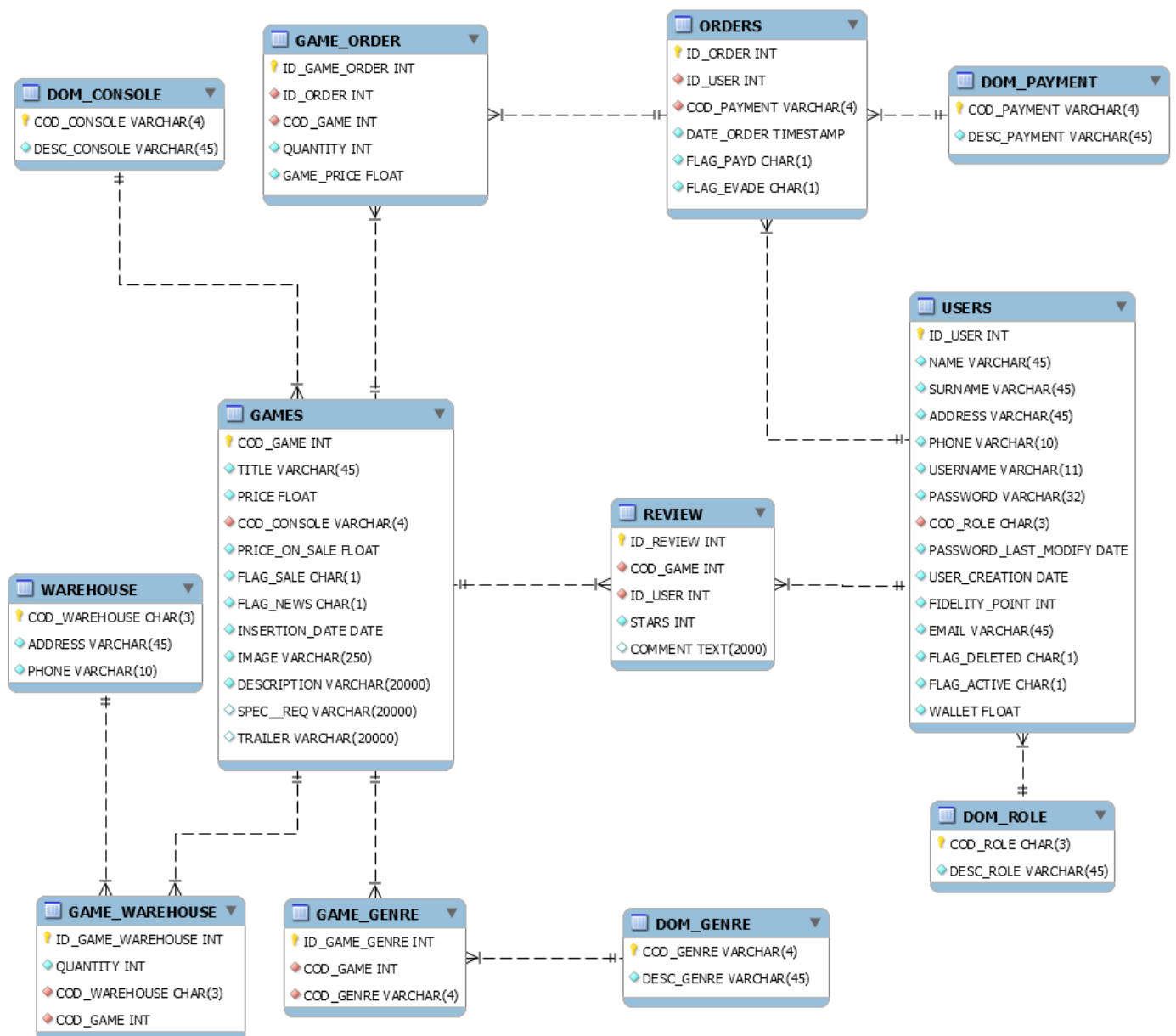
Per ogni ordine interessa sapere: la data dell'ordine, un controllo sullo stato di pagato, un controllo sullo stato di evaso, il tipo di pagamento scelto, la lista dei giochi presenti nell'ordine e le relative quantità, l'utente che ha effettuato l'ordine.

Per ogni utente interessa sapere: il nome, il cognome, l'indirizzo, il telefono, lo username, la password (criptata), la data di modifica della password, la data di creazione del profilo, l'email, il numero di punti della gestione fedeltà, il portafoglio, lo stato dell'account attivo e cancellato, il ruolo.

Inoltre ciascun utente può eseguire delle recensioni per ciascun gioco. Ogni recensione ha un testo e un voto da 1 a 5.

### 3.2 Schema Relazionale

Di seguito lo schema relazionale del database:



Come è possibile vedere dall'immagine, il modello concettuale è stato trasformato nel modello relazionale come segue.

Le relazioni n:m sono diventate nuove tabelle nel database aventi una chiave primaria come campo id e le due chiavi esterne relative alle due tabelle precedentemente collegate.

La password è stata criptata con codifica MD5 e viene gestita la conversione in lettura ed in inserimento.



### 3.3 Creazione del Database

Per la creazione del database è stato creato un file DLL contenente tutte le create necessarie:

```
-- DATABASE: NOPAYNOGAME
-- TABLE STRUCTURE FOR TABLE DOM_CONSOLE
CREATE TABLE DOM_CONSOLE (
  COD_CONSOLE VARCHAR(4) UNIQUE NOT NULL,
  DESC_CONSOLE VARCHAR(45) NOT NULL,

  CONSTRAINT DOM_CONSOLE_PK PRIMARY KEY (COD_CONSOLE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM_GENRE
CREATE TABLE DOM_GENRE (
  COD_GENRE VARCHAR(4) UNIQUE NOT NULL,
  DESC_GENRE VARCHAR(45) NOT NULL,

  CONSTRAINT DOM_GENRE_PK PRIMARY KEY (COD_GENRE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM_PAYMENT
CREATE TABLE DOM_PAYMENT (
  COD_PAYMENT VARCHAR(4) UNIQUE NOT NULL,
  DESC_PAYMENT VARCHAR(45) NOT NULL,

  CONSTRAINT DOM_PAYMENT_PK PRIMARY KEY (COD_PAYMENT)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM_ROLE
CREATE TABLE DOM_ROLE (
  COD_ROLE CHAR(3) UNIQUE NOT NULL,
  DESC_ROLE VARCHAR(45) NOT NULL,

  CONSTRAINT DOM_ROLE_PK PRIMARY KEY (COD_ROLE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAMES
CREATE TABLE GAMES (
  COD_GAME INT(11) UNIQUE NOT NULL AUTO_INCREMENT,
  TITLE VARCHAR(45) NOT NULL,
  PRICE FLOAT(11) NOT NULL,
  COD_CONSOLE VARCHAR(4) NOT NULL,
  PRICE_ON_SALE FLOAT(11) NOT NULL,
  FLAG_SALE CHAR(1) NOT NULL,
  FLAG_NEWS CHAR(1) DEFAULT 'Y' NOT NULL,

  IMAGE VARCHAR(250) DEFAULT 'http://placeholder.it/271x377' NOT NULL,
  DESCRIPTION VARCHAR(2000) NOT NULL,
  SPEC_REQ VARCHAR(2000),
  TRAILER VARCHAR(2000),

  INSERTION_DATE TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,

  CONSTRAINT GAMES_PK PRIMARY KEY (COD_GAME),

  CONSTRAINT GAMES_CK01 CHECK (FLAG_SALE IN ('Y', 'N')),
  CONSTRAINT GAMES_CK02 CHECK (FLAG_NEWS IN ('Y', 'N')),
  CONSTRAINT GAMES_CK03 CHECK (PRICE >= 0),
  CONSTRAINT GAMES_CK04 CHECK (PRICE_ON_SALE >= 0),

  CONSTRAINT GAMES_FK01 FOREIGN KEY (COD_CONSOLE) REFERENCES DOM_CONSOLE (COD_CONSOLE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME_GENRE
CREATE TABLE GAME_GENRE (
  ID_GAME_GENRE INT(11) NOT NULL AUTO_INCREMENT,
  COD_GAME INT(11) NOT NULL,
  COD_GENRE VARCHAR(4) NOT NULL,

  CONSTRAINT GAME_GENRE_PK PRIMARY KEY (ID_GAME_GENRE),

  CONSTRAINT GAME_GENRE_FK01 FOREIGN KEY (COD_GAME) REFERENCES GAMES (COD_GAME),
  CONSTRAINT GAME_GENRE_FK02 FOREIGN KEY (COD_GENRE) REFERENCES DOM_GENRE (COD_GENRE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE USERS
CREATE TABLE USERS (
  ID_USER INT(11) NOT NULL AUTO_INCREMENT,
  NAME VARCHAR(45) NOT NULL,
  SURNAME VARCHAR(45) NOT NULL,
  ADDRESS VARCHAR(45) NOT NULL,
```

```

PHONE VARCHAR(10) NOT NULL,
USERNAME VARCHAR(11) UNIQUE NOT NULL,
PASSWORD VARCHAR(32) NOT NULL,
EMAIL VARCHAR(45) UNIQUE NOT NULL,
COD_ROLE CHAR(3) NOT NULL,
PASSWORD_LAST_MODIFY TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
FIDELITY_POINT INT(11) DEFAULT 0 NOT NULL,
FLAG_DELETED CHAR(1) DEFAULT 'N' NOT NULL,
FLAG_ACTIVE CHAR(1) DEFAULT 'N' NOT NULL,

CONSTRAINT USERS_PK PRIMARY KEY (ID_USER),

CONSTRAINT USERS_FK01 FOREIGN KEY (COD_ROLE) REFERENCES DOM_ROLE(COD_ROLE),

CONSTRAINT USERS_CK01 CHECK (FIDELITY_POINT >= 0),
CONSTRAINT USERS_CK02 CHECK (FLAG_DELETED IN ('Y', 'N')),
CONSTRAINT USERS_CK03 CHECK (FLAG_ACTIVE IN ('Y', 'N'))
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE ORDERS
CREATE TABLE ORDERS (
ID_ORDER INT(11) NOT NULL AUTO INCREMENT,
ID_USER INT(11) NOT NULL,
COD_PAYMENT VARCHAR(4) NOT NULL,
DATE_ORDER TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
FLAG_PAYD CHAR(1) DEFAULT 'N' NOT NULL,
FLAG_EVADE CHAR(1) DEFAULT 'N' NOT NULL,

CONSTRAINT ORDERS_PK PRIMARY KEY (ID_ORDER),

CONSTRAINT ORDERS_FK01 FOREIGN KEY (ID_USER) REFERENCES USERS(ID_USER),
CONSTRAINT ORDERS_FK02 FOREIGN KEY (COD_PAYMENT) REFERENCES DOM_PAYMENT(COD_PAYMENT),
CONSTRAINT ORDERS_CK01 CHECK (FLAG_PAYD IN ('Y', 'N')),
CONSTRAINT ORDERS_CK02 CHECK (FLAG_EVADE IN ('Y', 'N'))
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME_ORDER
CREATE TABLE GAME_ORDER (
ID_GAME_ORDER INT(11) NOT NULL AUTO_INCREMENT,
ID_ORDER INT(11) NOT NULL,
COD_GAME INT(11) NOT NULL,
QUANTITY INT(11) DEFAULT 0 NOT NULL,
GAME_PRICE FLOAT(11) NOT NULL,

CONSTRAINT GAME_ORDER_PK PRIMARY KEY (ID_GAME_ORDER),

CONSTRAINT GAME_ORDER_FK01 FOREIGN KEY (COD_GAME) REFERENCES GAMES(COD_GAME),
CONSTRAINT GAME_ORDER_FK02 FOREIGN KEY (ID_ORDER) REFERENCES ORDERS(ID_ORDER),

CONSTRAINT GAME_ORDER_CK01 CHECK (QUANTITY >= 0)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE REVIEW
CREATE TABLE REVIEW (
ID_REVIEW INT(11) NOT NULL AUTO_INCREMENT,
COD_GAME INT(11) NOT NULL,
ID_USER INT(11) NOT NULL,
STARS INT(11) NOT NULL,
COMMENT_TEXT VARCHAR(2000),

CONSTRAINT REVIEW_PK PRIMARY KEY (ID_REVIEW),

CONSTRAINT REVIEW_FK01 FOREIGN KEY (COD_GAME) REFERENCES GAMES(COD_GAME),
CONSTRAINT REVIEW_FK02 FOREIGN KEY (ID_USER) REFERENCES USERS(ID_USER),

CONSTRAINT REVIEW_CK01 CHECK (STARS BETWEEN 1 AND 5)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE WAREHOUSE
CREATE TABLE WAREHOUSE (
COD_WAREHOUSE CHAR(3) NOT NULL,
ADDRESS VARCHAR(45) NOT NULL,
PHONE VARCHAR(45) NOT NULL,

CONSTRAINT WAREHOUSE_PK PRIMARY KEY (COD_WAREHOUSE)
) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME_WAREHOUSE
CREATE TABLE GAME_WAREHOUSE (
ID_GAME_WAREHOUSE INT(11) NOT NULL AUTO INCREMENT,
QUANTITY INT(11) NOT NULL,
COD_WAREHOUSE CHAR(3) NOT NULL,

```

```
COD_GAME INT(11) NOT NULL,  
  
CONSTRAINT GAME_WAREHOUSE_PK PRIMARY KEY (ID_GAME_WAREHOUSE),  
  
CONSTRAINT GAME_WAREHOUSE_FK01 FOREIGN KEY (COD_WAREHOUSE) REFERENCES WAREHOUSE(COD_WAREHOUSE),  
CONSTRAINT GAME_WAREHOUSE_FK02 FOREIGN KEY (COD_GAME) REFERENCES GAMES(COD_GAME),  
  
CONSTRAINT GAME_WAREHOUSE_CK01 CHECK (QUANTITY >= 0)  
) ENGINE=InnoDB;
```

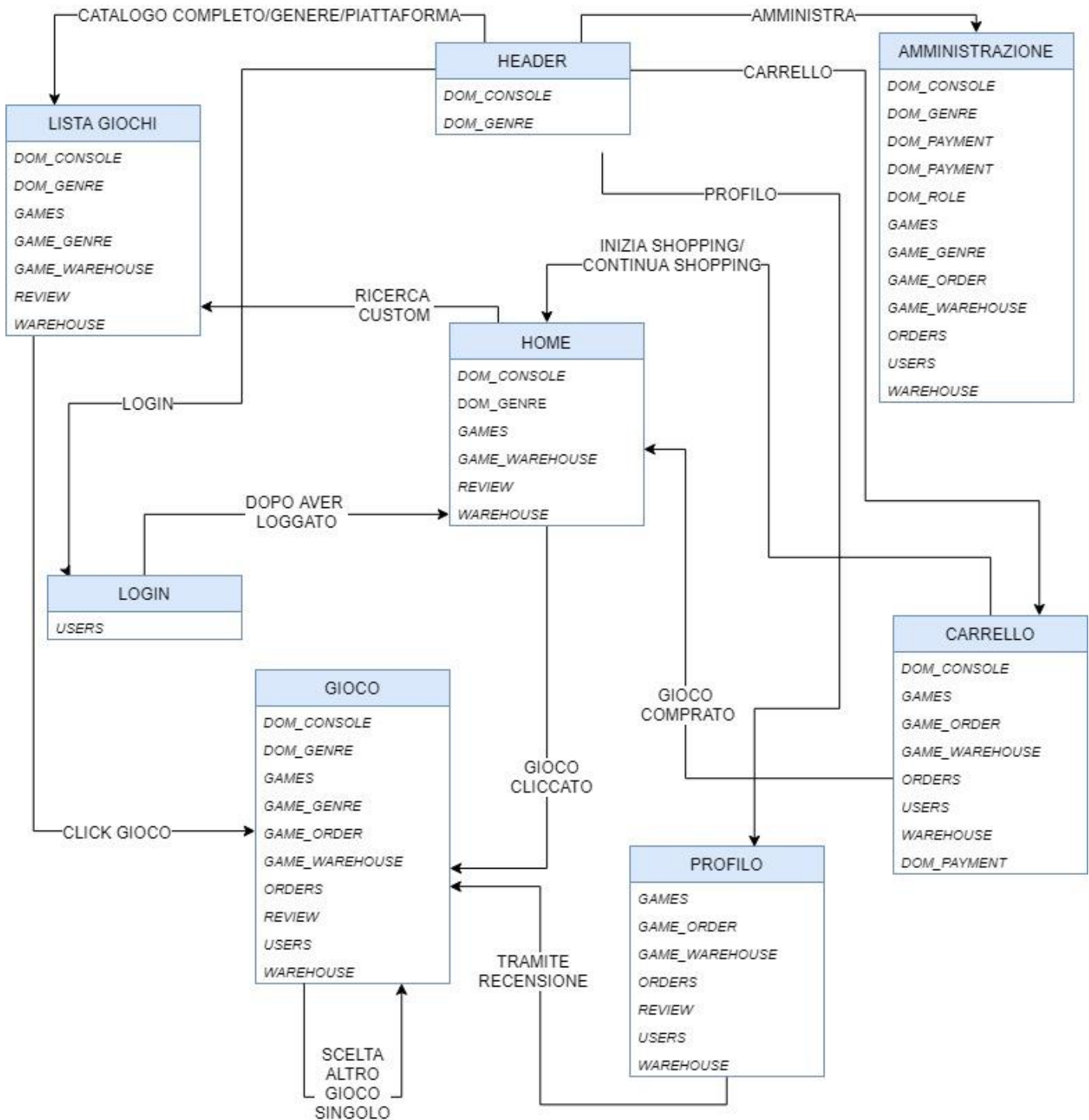
### 3.4 Popolamento del Database

Per il popolamento del database è stato creato un file DML con le insert necessarie per il popolamento delle tabelle di dominio al fine di garantire un corretto avvio del sito web:

Il resto del database è stato popolato dalla registrazione di utenti fisici e dall'immissione dei giochi dalla sezione di amministrazione.

## 4 Diagramma Applicazione Web

Di seguito viene riportato il diagramma dell'applicazione.



Il diagramma rappresenta il flusso del sito. Per evitare troppa confusione all'interno del diagramma è stata inserita una tabella "header" collegata a tutte le possibili pagine, poiché si trova nella parte superiore di tutte le pagine.

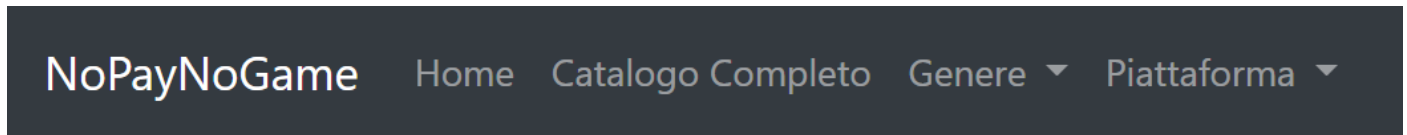
Ogni tabella è composta dal nome della pagina (con background blu) e una lista contenente i nomi delle rispettive tabelle contenute nel database e che vengono utilizzate all'interno di quella pagina.

Le frecce rappresentano lo spostamento tra le pagine e le descrizioni sulle frecce indicano l'evento scatenante (e.g. utilizzando un bottone o conseguentemente ad una azione).

## 5 Interfacce

Di seguito sono riportate le interfacce dell'applicazione web con una breve spiegazione delle query utilizzate per ciascuna interfaccia.

### 5.1 Header

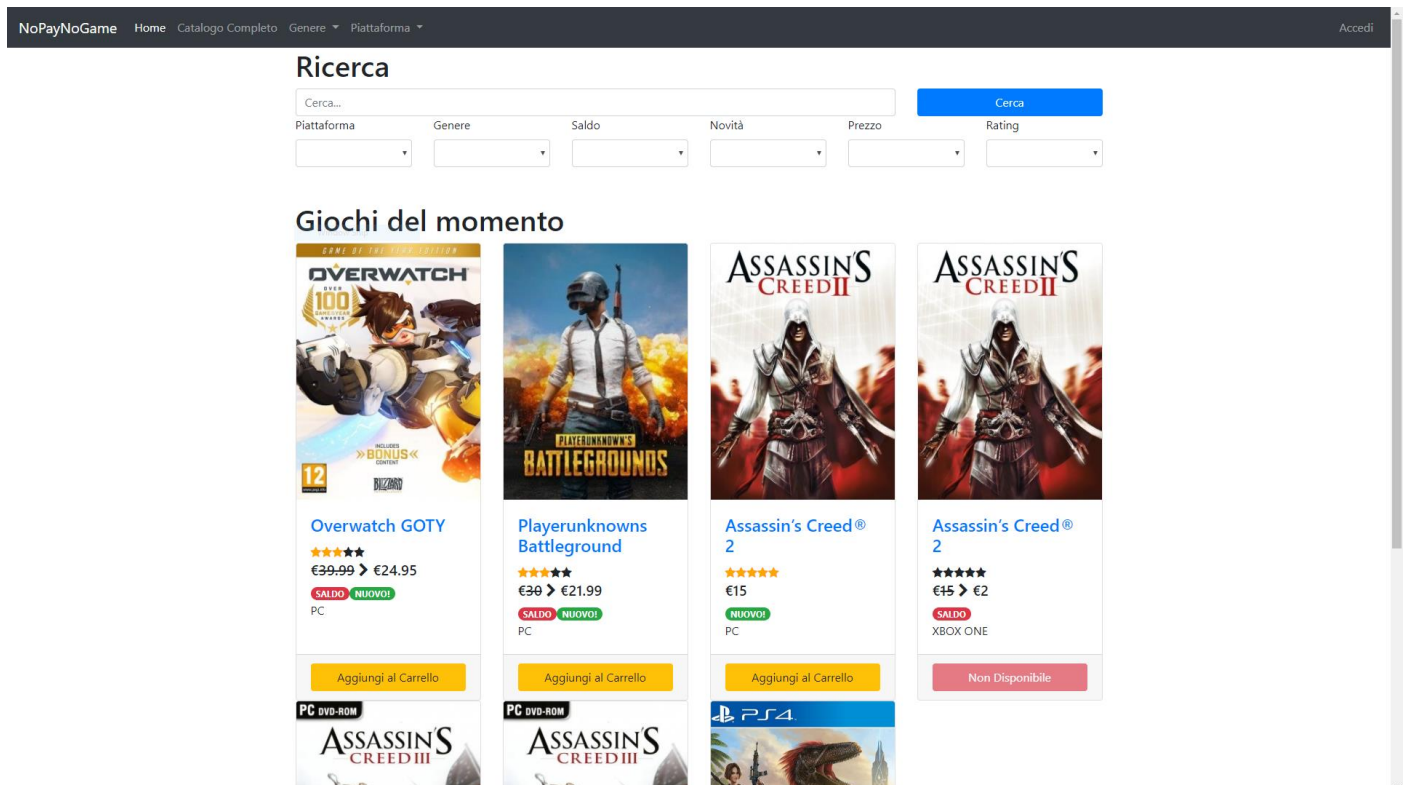


Per popolare l'header sono state fatte le seguenti query:

```
"SELECT * FROM my_nopaynogame.DOM_GENRE"  
"SELECT * FROM my_nopaynogame.DOM_CONSOLE"
```

I risultati ottenuti sono stati scorsi con un ciclo per poter popolare i 2 dropdown menu.

### 5.2 Home



L'home mostra sempre l'elenco dei giochi in saldo o le novità. Nella parte superiore è presente un form di ricerca che è stato popolato come segue:

- Piattaforma:

```
"SELECT * FROM my_nopaynogame.DOM_CONSOLE"
```

- Genere:

```
"SELECT * FROM my_nopaynogame.DOM_GENRE"
```

- Saldo:

```
<option value="Y">SI</option>  
<option value="N">NO</option>
```

- Novità:

```
<option value="Y">SI</option>  
<option value="N">NO</option>
```

- Prezzo:

```
<option value="0">0 - 15 €</option>  
<option value="15">15 - 30 €</option>
```

```
<option value="30">30 - 45 €</option>
<option value="45">45 - 60 €</option>
<option value="60">60 - 75 €</option>
```

- Rating:

```
<option value="-1" selected></option>
<option value="0">0+ stelle</option>
<option value="1">1+ stelle</option>
<option value="2">2+ stelle</option>
<option value="3">3+ stelle</option>
<option value="4">4+ stelle</option>
```


Per popolare la pagina dell'home sono state fatte le seguenti query:

```
"SELECT * FROM my_nopaynogame.GAMES WHERE flag_sale = 'Y' or flag_news = 'Y' "
```

### 5.2.1 Genere

NoPayNoGame
Home
Catalogo Completo
Genere ▼
Piattaforma ▼
Accedi

#### Genere AZIONE



Playerunknown's Battleground


★★★★★

€30 > €21.99

**SALDO** **NUOVO!**

PC

Aggiungi al Carrello



Assassin's Creed® 2

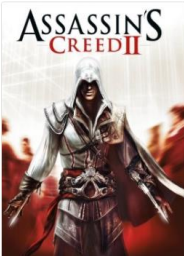
★★★★★

€15

**NUOVO!**

PC

Aggiungi al Carrello



Assassin's Creed® 2

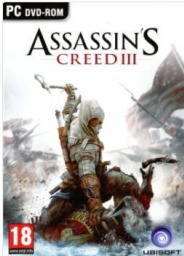
★★★★★

€45 > €2

**SALDO**

XBOX ONE

Non Disponibile



Assassins Creed III

★★★★★

€20 > €4.44

**SALDO**

PC

Aggiungi al Carrello

Il risultato scelto un genere è generato dalle seguenti query:

- Elenco Giochi:

```
"SELECT g.* FROM my_nopaynogame.GAMES g, my_nopaynogame.GAME_GENRE gg
WHERE gg.cod_genre = '$id' and g.cod_game = gg.cod_game"
```

- Nome genere scelto:

```
SELECT desc_genre FROM my_nopaynogame.DOM_GENRE WHERE cod_genre = '$id'"
```

### 5.2.2 Console

NoPayNoGame
Home
Catalogo Completo
Genere ▼
Piattaforma ▼
Accedi

#### Piattaforma XBOX ONE X

Nessun risultato trovato.

Il risultato scelto una piattaforma è generato dalle seguenti query:

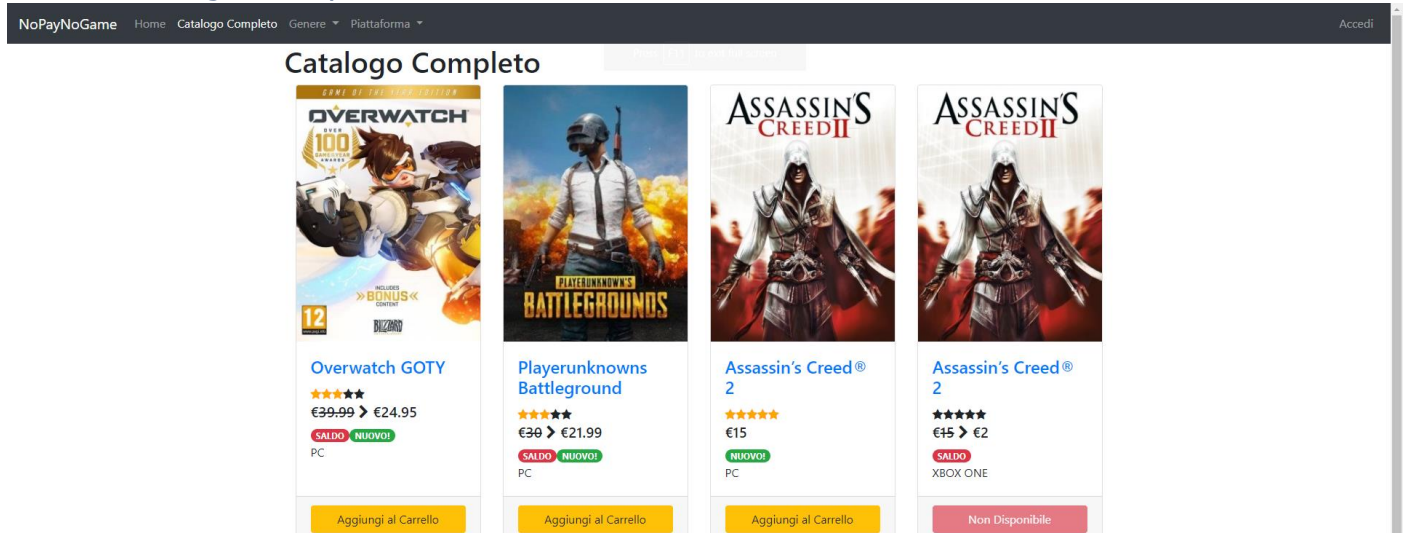
- Elenco Giochi:

```
"SELECT * FROM my_nopaynogame.GAMES WHERE cod_console = '$id'"
```

- Nome piattaforma scelta:

```
"SELECT desc_console FROM my_nopaynogame.DOM_CONSOLE WHERE cod_console = '$id'"
```

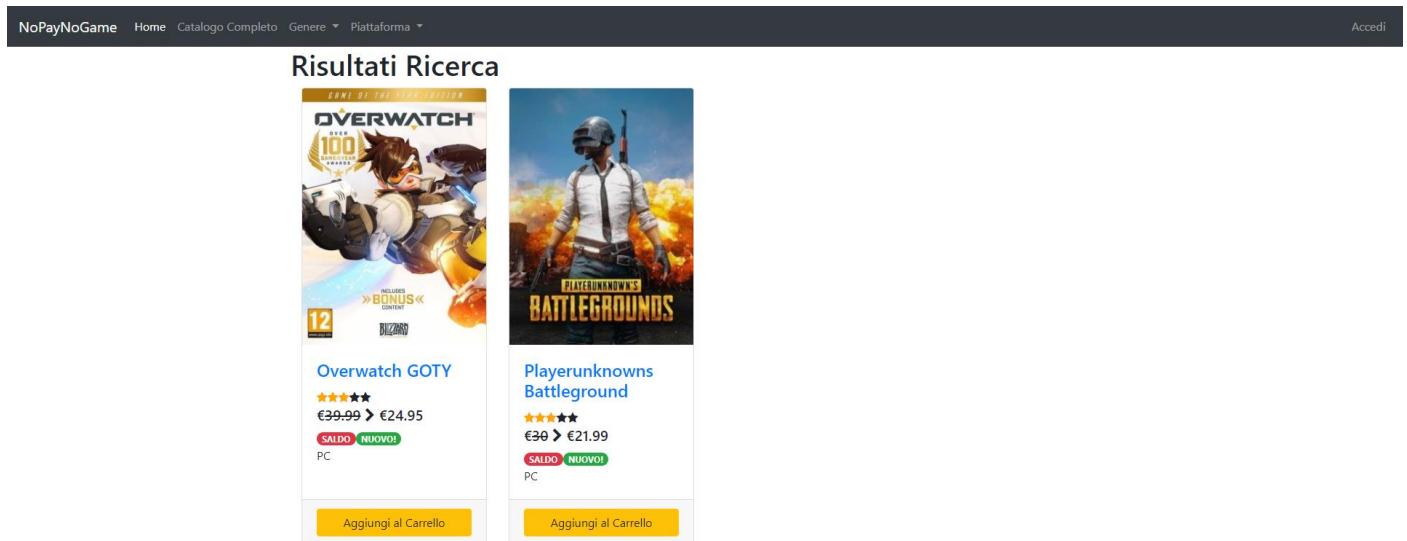
## 5.2.3 Catalogo Completo



Il risultato del catalogo completo è generato dalla seguente:

```
"SELECT * FROM my_nopaynogame.GAMES"
```

## 5.2.4 Ricerca



Il risultato della ricerca è generato dalla seguente query:

```
SELECT g.* FROM my_nopaynogame.GAMES g LEFT JOIN my_nopaynogame.GAME_GENRE gg
ON g.cod_game = gg.cod_game LEFT JOIN my_nopaynogame.REVIEW r ON g.cod_game = r.cod_game
WHERE g.title like '%"$.cod_t.%'
AND g.flag_news like '%"$.cod_n.%'
AND g.flag_sale like '%"$.cod_s.%'
AND (gg.cod_genre like '%"$.cod_g.%' OR gg.cod_genre IS NULL)
AND g.cod_console like '%"$.cod_c.%'
AND g.price_on_sale BETWEEN $cod_p $AND $cod_p+$range
GROUP BY g.cod_game
HAVING (AVG(r.stars) >= $cod_r) or ($cod_r < 1)"
```




## 5.3 Gioco

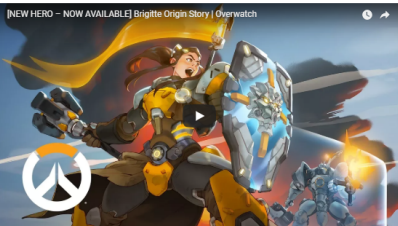
NoPayNoGameHomeCatalogo CompletoGenere \*Piattaforma \*

CarrelloRiesca di campi \*

### Overwatch GOTY



OVERWATCH  
100  
12  
★★★★★ 5.5  
€39.99 > €24.95  
SALDO  
BLUETOOTH  
PC  
Aggiungi al Carrello  
UNASINCRONIZZATO - FPS - MIMO - MOBA



NEW HERO - NOW AVAILABLE! Brigitte Origin Story | Overwatch

Descrizione

Overwatch BEST GAME

Requisiti di sistema

OS Win 7 64 CPU Core i3-540 3.06GHz / Phenom II X3 720  
RAM 4 GB  
GPU GeForce GTX 460 / Radeon HD 4850  
HDD 5 GB

Recensioni

a  
Questo gioco è davvero bellissimo, non so come ho fatto a vivere senza. A voi piace? ACQUISTO VERIFICATO ★★★★★

Admin  
Tascottfasp a atgufa gfas gsof g ★★★★★





Recensione

★★★★★ ★★★★★ ★★★★★ ★★★★★ ★★★★★

Scrivi qui il tuo commento

Invia

Giochi consigliati



Il risultato della seguente query seleziona tutte le informazioni inerenti al determinato gioco passato tramite \$codice\_gioco.

```
"SELECT * FROM GAMES WHERE COD_GAME ='$codice_gioco' "
```



### 5.3.1 Card Gioco



Di seguito viene fornita una descrizione per ciascuna parte della card:

- Stelline: la seguente query ricava la media delle stelle presente in review, è servito l'utilizzo di decimal per limitare il numero di cifre dopo la virgola.

```
"SELECT cast(AVG(stars)AS DECIMAL(10,1)) FROM my_nopaynogame.REVIEW WHERE  
od_game='$Cod_gioco'"
```

- Piattaforma: la seguente query ricava l'informazione della console appartenente al gioco

```
SELECT * FROM DOM_CONSOLE WHERE COD_CONSOLE = '$Cod_console'
```

- Bottone carrello: la seguente query serve per il carrello

```
echo "<form method='POST' action='addtocart.php'>  
<input type='hidden' name='cod_gioco' value='.$Cod_gioco.' />";  
$quantita = mysql_fetch_row(mysql_query("  
SELECT SUM(quantity) FROM my_nopaynogame.GAME_WAREHOUSE WHERE cod_game='$Cod_gioco'  
"));  
echo"<button type='submit' class='btn btn-block ";  
if($quantita[0] == 0){  
    echo"btn-danger' disabled>Non Disponibile</button>";  
}else{
```

```
echo"btn-warning">Aggiungi al Carrello</button>";
}
echo "</form>";
```


- Nome/Prezzo/Badge: La seguente query ricava tutte le informazioni del gioco per poi essere utilizzate per creare la descrizione del progetto

```
"SELECT * FROM GAMES WHERE COD_GAME = '$codice_gioco' "
```

### 5.3.2 Descrizione

NoPayNoGame
Home
Catalogo Completo
Genere
Piattaforma
Carrello
Alessia di campi

## Overwatch GOTY



OVERWATCH  
100  
ANNIVERSARY

12

★★★★★ 3.5

€39.99 > €24.95

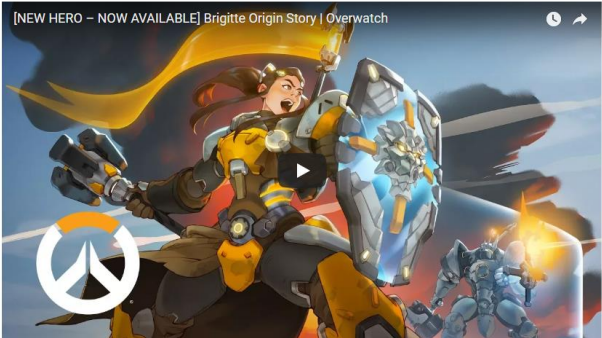
**NUOVO!**

PC

Aggiungi al Carrello

SPARATUTTO FPS MMO MOBA

[NEW HERO – NOW AVAILABLE] Brigitte Origin Story | Overwatch



Descrizione

Overwatch BEST GAME

Requisiti di sistema

OS Win 7 64 CPU Core i3-540 3.06GHz / Phenom II X3 720  
RAM 4 GB  
GPU GeForce GTX 460 / Radeon HD 4850  
HDD 5 GB

La descrizione, il trailer, l'elenco dei requisiti è stato popolato grazie alla query fatta in creazione di pagina al paragrafo Giuco.

### 5.3.3 Recensioni

Di seguito viene fornita una descrizione per ciascuna parte della recensione:

- Recensione: la seguente query è stata utilizzata per ricavare tutte le recensioni del gioco

```
"SELECT * FROM REVIEW WHERE COD_GAME = '$codice_gioco' "
```

- Utente recensione: la seguente query serve per ricavare le informazioni dell'utente che ha fatto la recensione

```
"SELECT * FROM USERS WHERE ID_USER = '$Id_user' "
```

#### 5.3.3.1 Loggato

##### Recensioni

a
ACQUISTO VERIFICATO ★★★★★
Questo gioco è davvero bellissimo, non so come ho fatto a vivere senza. A voi piace?

Admin
★★★★★
1sadfdasg q dfgsfa gfdg gsdg g

##### Recensione

★ ★ ★ ★ ★

Scrivi qui il tuo commento

Invia

La seguente query è stata utilizzata per controllare se un utente ha già effettuato una recensione per quel determinato gioco, evitando così che possa fare più recensioni dello stesso gioco

```
$query_controllo_commento = "SELECT * FROM REVIEW re, USERS usr WHERE re.ID_USER =  
usr.ID_USER AND re.ID_USER = '$Id_utente_cod' AND re.COD_GAME = '$Cod_gioco'";
```

La seguente insert serve per inserire la recensione all'interno del database nel caso in cui sia possibile farlo.

```
$query_insert="INSERT INTO `REVIEW`(`COD_GAME`, `ID_USER`, `STARS`, `COMMENT_TEXT`)  
VALUES ('$Cod_gioco', '$Id_utente_cod', '$Stella', '$commento')";
```

#### 5.3.3.2 Non Loggato

##### Recensioni

a

Questo gioco è davvero bellissimo, non so come ho fatto a vivere senza. A voi piace?

ACQUISTO VERIFICATO ★★★★★

Admin

1sdfdfasg q dfgsa gfdg sdf g

★★★★★

Devi essere loggato per poter lasciare una recensione

Loggati

La seguente query serve per controllare che un utente sia loggato per poter inserire recensioni.

```
$query_codice_utente = "SELECT ID_USER FROM USERS WHERE USERNAME = '$utente' OR EMAIL =  
'$utente'";
```

#### 5.3.3.3 Acquisto Verificato

a

dd

ACQUISTO VERIFICATO ★★★★★

La seguente query controlla se l'utente della recensione ha effettivamente comprato il gioco così da poter segnalare l'acquisto verificato.

```
$query_review_verificato = "SELECT * FROM USERS us, GAME_ORDER go, ORDERS ord WHERE  
us.ID_USER = ord.ID_USER AND ord.ID_ORDER = go.ID_ORDER AND ord.ID_USER = '$utente'";
```

#### 5.3.4 Giochi Consigliati

##### Giochi consigliati



La seguente query serve per popolare il carosello posto a fine pagina e che mostra dei giochi consigliati casuali della stessa piattaforma del gioco consigliato e lo stesso genere.

```
SELECT *
FROM GAMES g JOIN GAME_GENRE gg ON g.COD_GAME = gg.COD_GAME
WHERE g.COD_CONSOLE = '$Cod_console'
AND gg.COD_GENRE in (
    SELECT gg2.COD_GENRE
    FROM GAME_GENRE gg2
    WHERE gg2.COD_GAME = '$Cod_gioco')
ORDER BY RAND() LIMIT 8
```

## 5.4 Log In

NoPayNoGame

Home

Catalogo Completo

Genere

Piattaforma

Accedi

Accedi:

Email o Username

Enter email or your Username

Non condividiamo la tua email con nessuno.

Password

Password

[Hai dimenticato l'email?](#)

[Hai dimenticato password?](#)

Accedi

Registrati:

Nome

Mario

Cognome

Rossi

Email

Email

Password

Password

Username

MarioRossi

Indirizzo

1234 Main St

Telefono

3294252886

Registrati

### 5.4.1 Accesso

La seguente query è stata utilizzata nel momento in cui un utente accede e controlla che l'account non sia stato eliminato dal database

```
SELECT cod_role,id_user,name,surname FROM USERS WHERE PASSWORD='$passworda' &&
(EMAIL='$emaila' || USERNAME='$emaila') && FLAG_DELETED='N'
```

### 5.4.2 Registrazione

La seguente query viene controlla che l'email inserita non sia già stata utilizzata. Se l'utente esiste già non verrà inserito nuovamente ma gli verrà mostrato un alert il quale lo avviserà del fatto che possiede già un account attivo nel sito.

```
"SELECT * FROM USERS WHERE EMAIL='$email'"
```

La seguente insert viene utilizzata nel momento di avvenuta registrazione, inserendo tutte le informazioni nella tabella USERS.

```
"INSERT INTO USERS(name, surname, address, phone, username, password, cod_role,email)
VALUES('$name','$surname','$address','$phone','$username','$password','$role','$email')"
```

## 5.5 Profilo

### 5.5.1 Informazioni

NoPayNoGame Home Catalogo Completo Genere Piattaforma Carrello Alessia di campi

### Il mio profilo

Ciao Alessia questo è il tuo profilo personale!

Portafoglio 0 Punti fedeltà 60

Informazioni Ordini Effettuati Review e Voti Il mio conto

Nome:	Alessia	✎
Cognome:	di campi	✎
Username:	a	
Password:	*****	✎
Indirizzo:	Via rossi 19 meolo 30020 VE	✎
Telefono:	4444444444	✎
Cancella Account		Elimina

La seguente query prende tutte le informazioni dell'utente \$utente connesso per ricavare tutte le informazioni del profilo da mostrare all'interno del profilo personale.

```
$utente=$_SESSION['user'];  
$query ="SELECT * FROM USERS WHERE EMAIL='$utente' || USERNAME='$utente'";
```

La seguente update serve per settare la flag delete a Y facendo risultare l'account come cancellato.

```
UPDATE USERS SET FLAG DELETED='Y' WHERE ID_USER = $utente
```

A questo punto l'utente non sarà più in grado di loggare nel sito ed usufruire dei vantaggi del profilo fintanto che un admin non riattivi il suo account.

#### 5.5.1.1 Modifica Dati

NoPayNoGame Home Catalogo Completo Genere Piattaforma Carrello Alessia di campi

### Modifica dati

Modifica Close

Informazioni Ordini Effettuati Review e Voti Il mio conto

Nome: Alessia

Tutte queste update servono per modificare i dati dell'utente loggato.

```
UPDATE USERS SET surname='$newcognome' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET NAME='$newname' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET username='$newusername' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET EMAIL='$newemail' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET address='$newaddress' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET phone='$newphone' WHERE USERNAME='$utente' || email='$utente'  
UPDATE USERS SET password='$newpassword' WHERE USERNAME='$utente' || email='$utente'
```

## 5.5.2 Ordini Effettuati

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

CarrelloAlessia di campi

Il mio profilo

Ciao Alessia questo è il tuo profilo personale!

Portafoglio0Punti fedeltà35

Informazioni

Ordini Effettuati

Review e Voti

Il mio conto

Legenda:

Giallo -> da spedire

Verde -> spedito

Numero ordine	Data	Paga	Elimina ordine	Visualizza e/o modifica ordine
115	2018-06-09 17:21:14	Pagato!		+
110	2018-06-09 13:56:34	Paga	Elimina ordine	+
109	2018-06-09 13:56:24	Pagato!		+
108	2018-06-09 13:56:08	Pagato!		+
107	2018-06-09 13:55:50	Paga		+

Numero ordine

Data

Paga

Elimina ordine

Visualizza e/o modifica ordine

105

2018-06-09 13:54:49

Pagato!

+

Numero ordine

Data

Paga

Elimina ordine

Visualizza e/o modifica ordine

104

2018-06-09 13:54:49

Pagato!

+

Numero ordine

Data

Paga

Elimina ordine

Visualizza e/o modifica ordine

103

2018-06-09 13:54:49

Pagato!

+

Pagamento 102

Metodo di pagamento

Bonifico

Paga

Close

La seguente update serve per settare l'ordine a pagato cioè pagamento effettuato.

```
UPDATE ORDERS SET FLAG_PAYD='Y' WHERE ID_ORDER='$pagamento'
```

La seguente query viene utilizzata per prendere tutti gli ordini effettuati dall'utente.

```
SELECT * FROM ORDERS WHERE ID_USER='$id_utente' ORDER BY ID_ORDER DESC
```

### 5.5.2.1 Visualizza Ordine Pagato

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

CarrelloAlessia di campi

Il mio profilo

Ciao Alessia questo è il tuo profilo personale!

Portafoglio0Punti fedeltà35

Informazioni

Ordini Effettuati

Review e Voti

Il mio conto

Legenda:

Giallo -> da spedire

Verde -> spedito

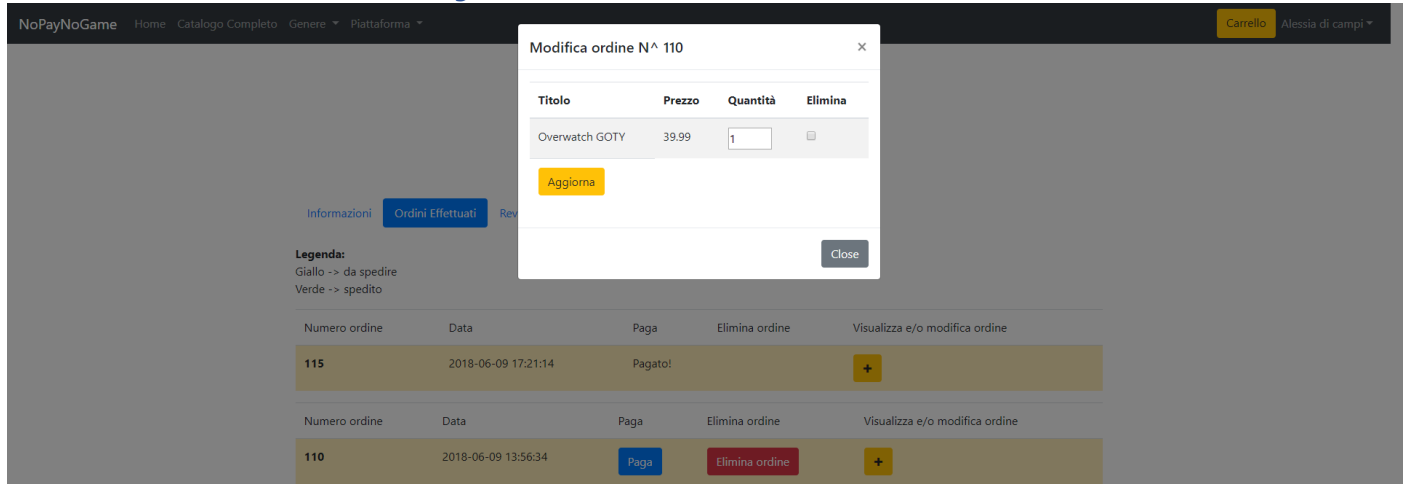
Titolo	Prezzo	Quantità	Elimina
Playerunknowns Battleground	30	1	X

Modifica ordine N^ 115

Aggiorna

Close

### 5.5.2.2 Visualizza Ordine Non Pagato



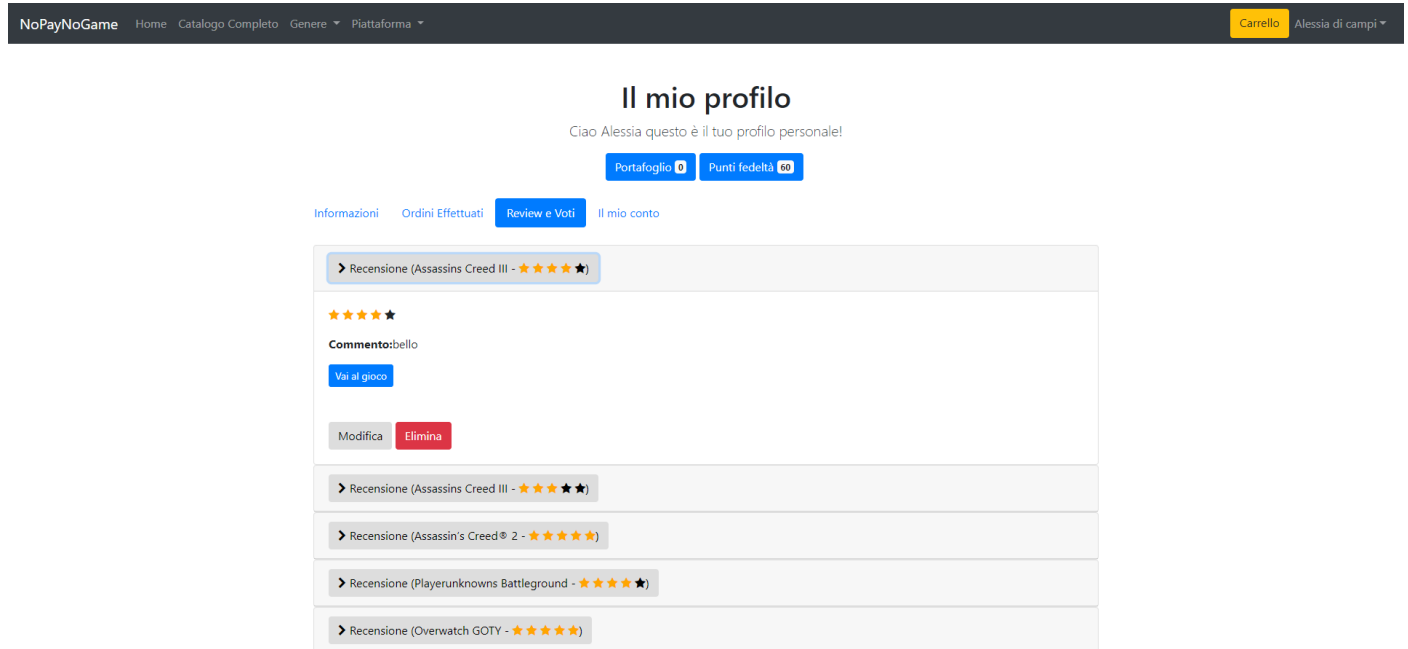
La seguente porzione di codice viene utilizzata al momento della cancellazione dell'ordine. Inizialmente vengono utilizzate delle query per prendere le informazioni inerenti al gioco ordinato e alla quantità contenuta in magazzino. Successivamente vengono rimessi i prodotti all'interno del warehouse e viene effettuata l'eliminazione dell'ordine.

```
$del_ordine = $_POST['del_ordine'];
$id_user = $_POST['id_user_order'];

$order_games= ("SELECT cod_game,quantity,game_price FROM GAME_ORDER WHERE id_order = '$del_ordine'");
$ris = ($conn->query($order_games));
foreach($ris as $riga){
    $cod_game_result = $riga['cod_game'];
    $quantity_result = $riga['quantity'];
    $game_price_result = $riga['game_price'];
    $warehouse = 'WH1';
    $quantity_warehouse= ("SELECT quantity FROM GAME_WAREHOUSE WHERE cod_game = '$cod_game_result'");
    $risultato = ($conn->query($quantity_warehouse));
    foreach($risultato as $rigas) {
        $safequantity = $rigas['quantity'];
    }
    $query = $conn -> prepare("UPDATE GAME_WAREHOUSE
SET
quantity = '$safequantity'+'$quantity_result'
WHERE cod_game = '$cod_game_result' and cod_warehouse = '$warehouse'");
    $query -> execute();
}
$query = $conn -> prepare("DELETE FROM GAME_ORDER WHERE id_order = '$del_ordine'");
$query -> execute();
$query = $conn -> prepare("DELETE FROM ORDERS WHERE id_order = '$del_ordine'");
$query -> execute();
```



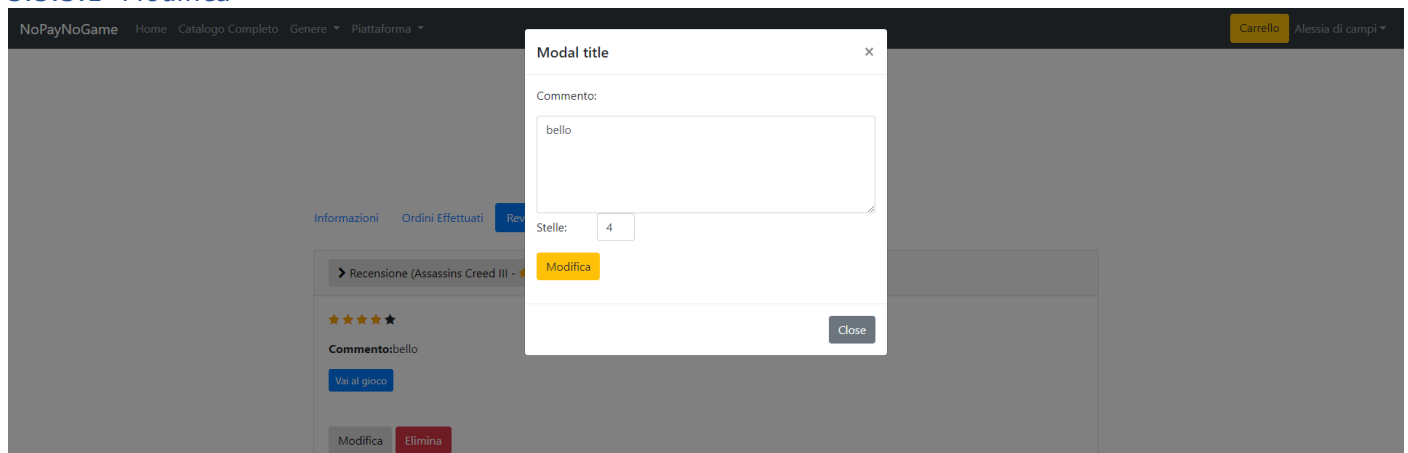
## 5.5.3 Review E Voti



La seguente query serve per prelevare tutte le recensioni fatte dall'utente.

```
SELECT * FROM REVIEW WHERE ID_USER='$id_utente' ORDER BY ID_REVIEW DESC
```

### 5.5.3.1 Modifica



La seguente delete serve per eliminare la recensione effettuata dall'utente.

```
DELETE FROM REVIEW WHERE ID_REVIEW='$rev'
```

La seguente update serve per modificare il commento e le stelle della recensione fatta su quel determinato \$modcommento.

```
UPDATE REVIEW SET COMMENT_TEXT='$testocommento' , STARS='$stelle' WHERE  
ID_REVIEW = '$modcommento'
```



## 5.5.4 Il Mio Conto

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

CarrelloAlessia di campi

### Il mio profilo

Ciao Alessia questo è il tuo profilo personale!

Portafoglio 0Punti fedeltà 20

InformazioniOrdini EffettuatiReview e VotiIl mio conto

100 punti

Con 100 punti potrai caricare il tuo portafoglio di 10 euro

10 euro

250 punti

Con 250 punti potrai caricare il tuo portafoglio di 30 euro

30 euro

500 punti

Con 500 punti potrai caricare il tuo portafoglio di 65 euro

65 euro

1000 punti

Con 1000 punti potrai caricare il tuo portafoglio di 140 euro

140 euro

La seguente update viene utilizzata nel caso in cui un utente decide di trasformare i propri punti fedeltà in soldi, andando così ad aggiornare il proprio portafoglio e il contatore dei punti fedeltà.

```
"UPDATE USERS SET WALLET=WALLET+'$ricarica' WHERE EMAIL='$utente' || USERNAME='$utente' "
"UPDATE USERS SET FIDELITY_POINT=FIDELITY_POINT-100 WHERE EMAIL='$utente' ||
USERNAME='$utente' "
```

## 5.6 Carrello

Tutti gli step del carrello vengono gestiti con un array salvato in \$SESSION.

### 5.6.1 Step 1 – Riepilogo Carrello

NoPayNoGame





HomeCatalogo CompletoGenerePiattaforma

Carrello 1Alessia di campi

### Carrello

Ciao Alessia questo è il tuo carrello

Portafoglio iniziale 0€

Prodotto	Prezzo	Quantità	Sub-totale		
PC	Playerunknowns Battleground	21.99	1	21.99	 
				21.99	 

Continue ShoppingCheckout

### 5.6.2 Step 2 - Spedizione

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

Carrello 1Alessia di campi

### Carrello

Ciao Alessia questo è il tuo carrello

Portafoglio iniziale 0€

Il tuo indirizzo:

Nome:

Alessia

Modifica

Cognome:

di campi

Modifica

Indirizzo:

Via rossi 19 meolo 30020 VE

Modifica

Torna indietro

Ultimo step!

### 5.6.3 Step 3 – Conferma Acquisto

NoPayNoGame Home Catalogo Completo Genere ▾ Piattaforma ▾ Carrello 1 Alessia di campi ▾

## Carrello

Ciao Alessia questo è il tuo carrello

Portafoglio iniziale €144.26€

Riepilogo prodotti:				Indirizzo di spedizione		
Titolo	Prezzo	Subtotale	Quantità	Nome	Cognome	Indirizzo
ARK Survival Evolved	50	50	1	Alessia	di campi	Via rossi 19 meolo 30020 VE

Torna indietro Invia ordine

### 5.6.4 Step 4 – Conferma Ricezione Ordine

NoPayNoGame Home Catalogo Completo Genere ▾ Piattaforma ▾ Carrello 1 Alessia di campi ▾


## Carrello

Ciao Alessia questo è il tuo carrello

Portafoglio iniziale €€

### Ordine ricevuto!

Id ordine: 79  
Saldo punti: 60  
Il tuo portafoglio: 0



Riceverai il tuo ordine tra 48h

La seguente query si occupa di inserire nel database un nuovo ordine non ancora pagato.

```
INSERT INTO `ORDERS` (`ID_USER`, `COD_PAYMENT`, `DATE_ORDER`, `FLAG_PAYD`, `FLAG_EVADE`)  
VALUES ('$ute', '$codepayment', '$data', 'N', 'N')
```

## 5.7 Amministrazione

### 5.7.1 Magazzino

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

AmministraCarrelloAdmin Ferroz

Amministrazione

MagazzinoAggiungi Nuovo GiocoGestione OrdiniGestione ProfiliAggiungi Profilo

ARK Survival Evolved - PS4 PRO - Copie attuali : 80

Assassin's Creed® 2 - XBOX ONE - Copie attuali : 0

Assassin's Creed® 2 - PC - Copie attuali : 210

Assassins Creed Black Flag - XBOX ONE S - Copie attuali : 155

Assassins Creed III - PS4 - Copie attuali : 50

Assassins Creed III - PC - Copie attuali : 800

Overwatch GOTY - PC - Copie attuali : 10081

Playerunknowns Battleground - PC - Copie attuali : 81

Questa query viene utilizzata nel pannello amministrazione per recuperare tutti i giochi e le relative quantità disponibili nei magazzini, ordinandoli per titolo.

```
SELECT g.*, SUM(gw.quantity) FROM my_nopaynogoame.GAMES g LEFT JOIN my_nopaynogoame.GAME_WAREHOUSE gw ON g.cod_game = gw.cod_game GROUP BY g.cod_game ORDER BY g.title"
```

Al momento dell'apertura di un singolo gioco è possibile eseguire una modifica dello stesso o una cancellazione.

NoPayNoGame

HomeCatalogo CompletoGenerePiattaforma

AmministraCarrelloAdmin Ferroz

Amministrazione

MagazzinoAggiungi Nuovo GiocoGestione OrdiniGestione ProfiliAggiungi Profilo

ARK Survival Evolved - PS4 PRO - Copie attuali : 80

Assassin's Creed® 2 - XBOX ONE - Copie attuali : 0

Assassin's Creed® 2 - PC - Copie attuali : 210

Assassins Creed Black Flag - XBOX ONE S - Copie attuali : 155

Assassins Creed III - PS4 - Copie attuali : 50

Assassins Creed III - PC - Copie attuali : 800

Overwatch GOTY - PC - Copie attuali : 10081

Titolo

Assassin's Creed® 2

Prezzo

15

Prezzo in Saldo

2

Novità

N

Console

XBOX ONE

Immagine

img/AC2.jpg

Descrizione

Assassin's Creed® 2 is the follow-up to the title that became the fastest-selling new IP in video game history. The highly

Requisiti Minimi

OS Win XP 32<br>CPU Core 2 Duo 1.8 GHZ / Athlon X2 64 2.4GHZ<br>

Trailer

AZIONE

AVVENTURA

SPORT

SPARATUTTO

FPS

MMO

RPG

PICCHIADURO

ARENA

MOBA

ROMPICAPO

PUZZLE

WH1 via Macello 1

0

WH2 via Budello 2

0

WH3 via Fretello 3

0

Aggiorna Gioco

Cancella Gioco

La seguente delete viene utilizzata per eliminare un gioco dal catalogo, è necessario eliminarlo dai giochi, dal magazzino e dalla lista dei giochi. Se una di queste delete non va a buon fine viene ripristinato.

```
try {  
    $conn -> beginTransaction();  
    $conn -> exec("DELETE FROM GAME_GENRE WHERE cod_game = '$modCodGame'");
```

```

$conn -> exec("DELETE FROM GAME_WAREHOUSE WHERE cod_game = '$modCodGame'");
$conn -> exec("DELETE FROM GAMES WHERE cod_game = '$modCodGame'");
$conn -> commit();

}catch (Exception $e){
    $conn -> rollBack();
}

```

La seguente update viene utilizzata per aggiornare tutte le informazioni legate ad un singolo gioco.

```

"UPDATE GAMES
SET
title = '$modTitle',
price = '$modPrice',
cod_console = '$modConsole',
price_on_sale = '$modPriceSale',
flag_sale = '$modSale',
flag_news = '$modNovita',
image = '$modImmagine',
description = '$modDesc',
spec_req = '$modReq',
trailer = '$modTrailer'
WHERE cod_game = '$modCodGame'"

```

Update che serve per aggiornare la quantità del gioco all'interno del magazzino.

```

"UPDATE GAME_WAREHOUSE
SET
quantity = '$quantita_magazzino'
WHERE cod_game = '$modCodGame' and cod_warehouse = '$cod_magazzino'"

```

Update che serve per aggiornare la lista dei generi del gioco.

```

"DELETE FROM GAME_GENRE WHERE cod_game = '$modCodGame'"
foreach($modGeneri as $co_genere) {
    $query = $conn -> prepare("INSERT INTO GAME_GENRE(cod_game, cod_genere)
VALUES('$modCodGame', '$co_genere')");
    $query -> execute();
}

```

## 5.7.2 Aggiunta Nuovo Gioco

NoPayNoGame
Home
Catalogo Completo
Genere
Piattaforma
Amministra
Carrello
Admin Ferroz

### Amministrare

Magazzino
Aggiungi Nuovo Gioco
Gestione Ordini
Gestione Profili
Aggiungi Profilo

Titolo	Prezzo	Prezzo in Saldo	Novità	Console	Immagine
<input type="text" value="Titolo"/>	<input type="text" value="xxx,xxx"/>	<input type="text" value="xxx,xxx"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="img/file.jpg"/>

Descrizione

Requisiti Minimi

Trailer

☐ AZIONE
☐ AVVENTURA
☐ SPORT
☐ SPARATUTTO
☐ FPS
☐ MMO
☐ RPG
☐ PICCHIADURO
☐ ARENA
☐ MOBA
☐ ROMPICAPO
☐ PUZZLE

WH1 via Macello 1

WH2 via Budello 2

WH3 via Fretello 3

Aggiungi

La seguente insert viene utilizzata per inserire all'interno del database tutte le informazioni del e le diverse quantità contenute nei singoli magazzini.

```

"INSERT INTO GAMES(title, price, cod_console, price_on_sale, flag_sale, flag_news, image,

```

```

description, spec_req,
trailer)VALUES('$title','$price','$cod_console','$price_on_sale','$flag_sale','$flag_news',
'$image','$description','$spec_req','$trailer')")

foreach($generi as $co_genere) {
    $query = $conn -> prepare("INSERT INTO GAME_GENRE(cod_game, cod_genre)
VALUES('$cod_game_result', '$co_genere')");
    $query -> execute();
}

foreach($magazzini as $cod_magazzino) {
    $quantita_magazzino = $quantita[$i];
    $query = $conn -> prepare("INSERT INTO GAME_WAREHOUSE(cod_game, cod_warehouse,
quantity)
VALUES('$cod_game_result', '$cod_magazzino', '$quantita_magazzino')");
    $query -> execute();
    $i = $i + 1;
}

```

### 5.7.3 Gestione Ordini

NoPayNoGame

Home

Catalogo Completo

Genere ▾

Piattaforma ▾

Amministra

Carrello

Admin Ferroz ▾

Amministrazione

Magazzino

Aggiungi Nuovo Gioco

Gestione Ordini

Gestione Profili

Aggiungi Profilo

Nr. Ordine	Username	Pagamento	Data	Nr. Articoli	Totale	Pagato?	Evaso?	Dettagli	Salva	Elimina
5	Chipmunk	PAYPAL	2018-04-17 22:20:18	15	254.84999656677246	<div><div>Y ▾</div></div>	<div><div>Y ▾</div></div>	<div>Vedi Dettagli</div>	<div>Salva</div>	<div>Elimina</div>

La seguente porzione di codice viene utilizzata nel momento della cancellazione dell'ordine. Inizialmente vengono utilizzate delle query per prendere le informazioni inerenti al gioco ordinato e alla quantità contenuta in magazzino.

Nel caso in cui venisse eliminato un ordine, vengono rimessi i prodotti all'interno del warehouse, viene rimborsato l'utente ed infine viene effettuata la delete dell'ordine.

La delete dell'ordine è possibile solamente nel caso in cui l'ordine non sia già stato pagato. L'evasione dell'ordine è possibile solamente ad ordine pagato.

```

/*QUERY PER ELIMINA ORDINE*/
$del_ordine = $_POST['del_ordine'];
$id_user = $_POST['id_user_order'];

$order_games= ("SELECT cod_game,quantity,game_price FROM GAME_ORDER WHERE id_order =
'$del_ordine'");
$ris = ($conn->query($order_games));
foreach($ris as $riga){
    $cod_game_result = $riga['cod_game'];
    $quantity_result = $riga['quantity'];
    $game_price_result = $riga['game_price'];
    $warehouse = 'WH1';

    $quantity_warehouse= ("SELECT quantity FROM GAME_WAREHOUSE WHERE cod_game =
'$cod_game_result'");
    $risultato = ($conn->query($quantity_warehouse));

```

```

foreach($risultato as $rigas) {
    $safequantity = $rigas['quantity'];
}

$query = $conn -> prepare("UPDATE GAME_WAREHOUSE
SET
quantity = '$safequantity'+'$quantity_result'
WHERE cod_game = '$cod_game_result' and cod_warehouse = '$warehouse'");
$query -> execute();

$quantity_warehouse= ("SELECT wallet FROM USERS WHERE id_user = '$id_user'");
$risultato = ($conn->query($quantity_warehouse));
foreach($risultato as $rigas) {
    $wallet = $rigas['wallet'];
}

$query = $conn -> prepare("UPDATE USERS
SET
wallet = '$wallet'+('$quantity_result'*'$game_price_result')
WHERE id_user = '$id_user'");
$query -> execute();
}

$query = $conn -> prepare("DELETE FROM GAME_ORDER WHERE id_order = '$del_ordine'");
$query -> execute();
$query = $conn -> prepare("DELETE FROM ORDERS WHERE id_order = '$del_ordine'");
$query -> execute();

```

Il seguente update viene utilizzato per segnalare con le diverse flag se l'ordine è stato pagato e/o evaso.

```

"UPDATE ORDERS SET flag_payd = '$flag_pagato', flag_evade = '$flag_evaso' WHERE id_order = '$del_ordine'"

```

## 5.7.4 Gestione Profili

NoPayNoGame Home Catalogo Completo Genere Piattaforma						Amministra Carrello Admin Ferroz
Amministrazione						
Magazzino	Aggiungi Nuovo Gioco	Gestione Ordini	Gestione Profili	Aggiungi Profilo		
Username	Ruolo	Punti	Attivo?	Cancellato?	Salva	
Admin	SUPERUSER	8900	Y	N	Salva	
alexis	USER	0	Y	N	Salva	
Chipmunk	SUPERUSER	0	Y	N	Salva	
roberta	AMMINISTRATORE	0	Y	N	Salva	
a	USER	0	N	N	Salva	
magazzino	MAGAZZINIERE	0	N	N	Salva	
FrancoB	USER	0	Y	N	Salva	
utente	USER	0	N	N	Salva	

Update per modificare le flag inerenti ai ruoli dell'utente e a segnalare se l'utente è stato cancellato e se è ancora attivo. L'attivazione è necessaria per poter utilizzare un nuovo profilo.

```

"UPDATE USERS SET cod_role = '$save_role', flag_deleted = '$save_deleted', flag_active = '$save_active' WHERE id_user = '$save_user'"

```

## 5.7.5 Aggiunta Nuovo Profilo

NoPayNoGameHomeCatalogo CompletoGenere ▾Piattaforma ▾

AmministraCarrelloAdmin Ferroz ▾

### Amministrazione

Magazzino

Aggiungi Nuovo Gioco

Gestione Ordini

Gestione Profili

Aggiungi Profilo

Nome	Cognome	Email	Password
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Username	Indirizzo	Phone	Ruolo
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Aggiungi

La seguente insert viene utilizzata al posto della registrazione creando un utente dall'amministratore. Inserisce tutti i campi come in fase di registrazione, con la possibilità ulteriore di scegliere un diverso ruolo.

```
"INSERT INTO USERS(name, surname, address, phone, username, password, cod_role, email)
VALUES('$name', '$surname', '$address', '$phone', '$username', '$password', '$role', '$email'
)"
```

## 6 Struttura del Progetto

img	03/06/2018 20:37	File folder	
account.php	03/06/2018 22:53	PHP File	40 KB
addtocart.php	03/06/2018 21:09	PHP File	1 KB
administrate.php	03/06/2018 22:07	PHP File	40 KB
checkout.php	03/06/2018 22:20	PHP File	4 KB
connection.php	03/06/2018 20:35	PHP File	1 KB
delete_account.php	03/06/2018 20:35	PHP File	1 KB
disconnection.php	03/06/2018 20:35	PHP File	1 KB
game.php	03/06/2018 22:47	PHP File	14 KB
gamelist.php	03/06/2018 22:42	PHP File	6 KB
header.php	03/06/2018 22:31	PHP File	5 KB
index.php	03/06/2018 22:28	PHP File	7 KB
script.php	03/06/2018 20:52	PHP File	1 KB
shoppingcart.php	03/06/2018 22:22	PHP File	25 KB
sign.php	11/06/2018 22:16	PHP File	8 KB
style	03/06/2018 20:45	Cascading Style S...	3 KB

L'immagine raffigura l'insieme dei file che sono stati creati per gestire l'applicazione web. Di seguito è riportata una breve spiegazione per ciascun file:

- img: cartella contenente tutte le immagini utilizzate;
- account.php: file per la creazione della pagina profilo utente;
- addtocart.php: script per la gestione del carrello;
- administrate.php: file per la creazione della pagina di amministrazione;
- checkout.php: script per la gestione dell'aggiunta di un ordine;
- connection.php: script per l'apertura della connessione con il database;
- delete\_account.php: script per la gestione della cancellazione logica di un account;
- disconnection.php: script per la chiusura della connessione con il database ed il drop della sessione corrente;
- game.php: file per la creazione della pagina del singolo gioco;
- gamelist.php: file per la creazione della pagina della lista dei giochi;
- header.php: file per la creazione dell'header;
- index.php: file per la creazione della pagina home;
- script.php: file per la gestione degli import dei vari script in tutte le pagine;
- shoppingcart.php: file per la creazione della pagina del carrello;
- sign.php: file per la creazione della pagina di login;
- style.css: file contenente alcuni stili css custom.