|  |
| --- |
| Università degli Studi di Salerno  Corso di Ingegneria del Software |

GameVault

Object Design Document

A logo for a video game

Description automatically generated

Team di Sviluppo:

|  |  |
| --- | --- |
| **Nome** | **Matricola** |
| Corsini Ernesto | 0512114376 |
| Orsatti Alessandro | 0512114982 |
| Palma Claudio Giuseppe | 0512114151 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autore** |
| 01/05/2024 | 1.0 | Prima stesura documento | Palma Claudio Giuseppe |
| 02/05/2024 | 1.1 | Completamento Prima Stesura | Palma Claudio Giuseppe |

|  |  |
| --- | --- |
| **Scritto da:** | Claudio Giuseppe Palma |

**Indice**

[1. INTRODUZIONE 3](#_Toc170060212)

[1.1 Object Design Trade-offs 3](#_Toc170060213)

[**1.1.1** **Robustezza vs Tempo** 3](#_Toc170060214)

[**1.1.2** **Attendibilità vs Tempo** 3](#_Toc170060215)

[1.2 Linee Guida 3](#_Toc170060216)

[1.3 Referenze 3](#_Toc170060217)

[2. DIRECTORY 3](#_Toc170060218)

[**2.1** **Back-End** 3](#_Toc170060219)

[**2.1.1** **webapp** 4](#_Toc170060220)

[**2.1.1.1** **META-INF** 4](#_Toc170060221)

[**2.1.1.2** **js** 4](#_Toc170060222)

[**2.1.1.3** **foto** 4](#_Toc170060223)

[**2.1.1.4** **html** 4](#_Toc170060224)

[**2.1.1.5** **css** 4](#_Toc170060225)

[**2.2** **db** 4](#_Toc170060226)

[3. PACCHETTI 4](#_Toc170060227)

[3.1 **BACK-END** 5](#_Toc170060228)

[4. INTERFACCE DI CLASSE 6](#_Toc170060229)

# INTRODUZIONE

## Object Design Trade-offs

### **Robustezza vs Tempo**

Il controllo dei dati un aspetto importante del sistema, farlo però nella maniera più corretta e completa possibile richiederebbe un tempo di sviluppo maggiore rispetto a quello a nostra disposizione per il rilascio di una nuova versione completamente funzionante. A discapito della robustezza, dunque, decidiamo di limitare i controlli sui dati in input, cosa che però sarà fatta nelle versioni successive.

### **Attendibilità vs Tempo**

L’attenibilità è un requisito fondamentale per il funzionamento del sistema, è infatti, ad esempio, importante gestire le transazioni in maniera corretta evitando, che gli ordini possano essere effettuati senza avere la disponibilià del prodotto richiesto. Chiaramente questo requisito necessita di maggior tempo di sviluppo per essere implementato, ed è proprio grazie al tempo guadagnato a discapito della robustezza che possiamo rispettarlo.

## Linee Guida

Qui di seguito sono riportate alcune linee guida per la stesura del codice:

* Gli oggetti DAO devono essere nominati nomeentitaIDS.
* Gli errori devono essere gestiti tramite dei valori di ritorno.

## Referenze

* R.A.D.
* S.D.D.

# DIRECTORY

## **Back-End**

Questa directory contiene i file Php e JavaScript che compongono i codici del sistema.

### **webapp**

### **META-INF**

#### Questa directory contiene le meta-informazioni.

### **js**

#### Questa directory contiene i file JavaScript con numerose funzioni utilizzate per rendere il sito più fluido a livello di animazioni.

### **foto**

#### Questa directory contiene le foto utilizzare all’interno della web-app.

### **html**

#### Questa directory contiene i file html che rapprensentano il sito stesso.

### **css**

#### Questa directory contiene gli stili per il sito web e le formattazioni per il browser.

## **db**

Questa directory serve per mantenere lo schema del database relazionale.

# PACCHETTI

In questa sezione analizzeremo la suddivisione in pacchetti di tutte le classi implementate, ecco una prima panoramica del sistema implementato:

Several blue folders with different types of files

Description automatically generated

## **BACK-END**

Questo pacchetto contiene codici per la gestione dati e reindirizzamento, come verifica dei dati dal database, le varie operazioni eseguite dall’Admin oppure la ricerca di prodotti nel catalogo ed il conseguente reindirizzamento alla pagina del prodotto ricercato.

* + 1. **Sign-up**

Qui c’è il codice per la connessione al database in modo da registrare i dati ed effettuare il Sign-Up.

A blue sign up box

Description automatically generated

* + 1. **Log-In**

Qui c’è il codice per la connessione al database in modo da prelevare i dati ed effettuare il log-In.

A blue log in box

Description automatically generated

* + 1. **Admin PHP**

Qui c’è il codice che si occupa di comunicare con il database in base alle operazioni delll’Admin.

A blue card with black text

Description automatically generated

* + 1. **Catalogo**

Qui c’è il codice che si occupa di prelevare dal database tutti gli elementi della tabella catalogo e renderli visibli all’utente. Inoltre è possibile eseguire una ricerca specifica di un prodotto tramite il nome.

A blue box with black text

Description automatically generated

## **FRONT-END**

Questo pacchetto contiene codici per la visualizzazione delle pagine web.

A diagram of a computer flowchart

Description automatically generated

* + 1. **Adminweb**

Pagina accessibile solo all’Admin dove è possibile visualizzare e modificare i prodotti presenti sul sito e gli ordini effettuati.

* + 1. **Cart**

Pagina dove vengono visualizzati i prodotti aggiunti al carrello con l’opzione di modificare la quantità.

* + 1. **Catalog**

Pagina dove vengono visualizzati i prodotti acquistabili.

* + 1. **Home**

Pagina dove troviamo le ultime uscite, i giochi più popolari e accedere alla pagina del prodotto.

* + 1. **Login**

Pagina di accesso e registrazione all’area utente.

* + 1. **Product**

Pagina dove è possibile visualizzare nel dattaglio il prodotto ed eventualmente aggiungerlo al carrello.

* + 1. **Support**

Pagina dove è possibile contattare il nostro team, che si occupa di fornire assistenza per ogni problema riscontrato.

* + 1. **Transition**

Pagina dove si esegue il pagamneto dei prodotti del carrello.

# INTERFACCE DI CLASSE

4.1 Package Back-End

File signup.php e login-php

|  |  |
| --- | --- |
| Metodo | Descrizione |
| Sign-Up | Form che accetta in input dati di tipo type=”text”, li inserisce all’interno del DBMS e assegna ad ogni utente un id randomico. |
| Log-In | Form che accetta in input dati di tipo type=”text”, li confronta con quelli già presenti nel DBMS e se corrispondono permette l’accesso alla sua area personale, inizializzando la sessione. |

File prodotto.php

|  |  |
| --- | --- |
| Metodo | Descrizione |
| Ottenimento Dati | Richiesta di tipo POST al database he permette allo script di prendere i dati necessari dal DBMS. |
| Stampa | Fornisce l’output dei dati presi in input dal DBMS tramite una funzione echo. |

File admin.php

|  |  |
| --- | --- |
| Metodo | Descrizione |
| Aggiunta | Form che accetta in input dati di tipo type=”text” e type= “number”, e li va ad inserire in una nuova riga del DBMS, se non è già presente. |
| Modifica | Form che accetta in input dati di tipo type=”text” e type= “number”, e sovrascrive i vecchi dati all’interno del DBMS con i nuovi parametri inseriti precedentemente. |
| Rimozione | Form che accetta in input dati di tipo type= “number”, facendo riferimento all’id del gioco, controlla se presente nel database. Se presente tutta la riga viene eliminata. |

File transition.php

|  |  |
| --- | --- |
| Metodo | Descrizione |
| VerificaMeseAnno | Form che accetta in input dati di tipo type=”month”, ovvero date che contengono solo mese ed anno. Poi controlla se la carta utilizzata per il pagamento è scaduta. |

4.2 Package Front-End

File catalogo.php

|  |  |
| --- | --- |
| Metodo | Descrizione |
| StampaTutto | Esegue una stampa di tutti i prodotti inseriti nel DBMS. |
| RicercaProdotto | Form che accetta in input dati di tipo type=”text”, li confronta con quelli già presenti nel DBMS e se corrispondono, anche parzialmente, vengono stampati nella pagina del catalogo. (Es. Se nella ricerca si inserisce “Metal Gear” il sistema stamperà tutti i prodotti che contengono anche parzialmente quel testo; come “Metal gear”, “Metal gear 2”, “Metal gear Solid” ecc…) |