|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ODD  Progetto  STRAGAME   |  |  | | --- | --- | | Riferimento |  | | Versione | 0.1 | | Data | 16/11/2020 | | Destinatario | Prof.ssa F. Ferrucci | | Presentato da | STRANIVERSITY (Umberto Mauro, Vincenzo Iovino, Mario Maffettone, Carmine Laudato) | | Approvato da |  | |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 12/12/2020 | 0.1 | Primo capitolo e inizio del secondo capitolo | Vincenzo Iovino  Carmine Laudato |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Indice

1. Introduzione 4
   1. Object Design trade-off 4
   2. Interface Documentation Guidelines 4
      1. Package 5
      2. Pagine HTML 5
      3. Fogli di stile CSS 5
      4. Script JavaScript e JSP Indentazione 5
      5. Nomi 5
      6. Posizione dichiarazione variabili 5
      7. Database SQL 6
      8. Organizzazione dei file 6
   3. Definizioni, Acronimi e Abbreviazioni 6
   4. Riferimenti 7
2. Packages 7
   1. View 7
   2. Model 9
   3. Controller 4
3. Class Interfaces 4
4. Design Pattern con Class Diagram 4
   1. Data Access Object Pattern 4
   2. Facade Pattern 4
5. Glossario 4
6. **Introduzione**
   1. **Object Design trade-off**

Dopo la realizzazione del documento RAD (Requirement Analysis Document) e SDD (System Design Document), abbiamo descritto in linea di massima, quello che sarà il nostro sistema e quindi i nostri obiettivi, tralasciando gli aspetti implementativi.

* **Prestazioni VS Costi**

Considerando il sistema che stiamo realizzando, possiamo dire che il non eccessivo budget a nostra disposizione ci ha consentito di realizzare il prodotto utilizzando materiale open source partendo da zero minimizzando i costi e rendendo il sistema più che soddisfacente.

* **Interfaccia VS Tempo di risposta**

Il tempo di risposta tra server e interfaccia sono più che sufficienti(rapidi), a soddisfare le esigenze dei vari videogiocatori collegati al sistema.

* **Interfaccia VS Easy-use**

L’interfaccia, grazie all’utilizzo delle form e di una impostazione semplice e intuitiva, permette un suo facile (Easy-Use) della gestione del sistema di database, permettendo l’immediata attività anche ai meno esperti col computer.

* **Costi VS Mantenimento**

Grazie a un uso di materiale open source e l’utilizzo del linguaggio javadoc, il sistema può essere facilmente modificato, aggiungendo nuove funzioni o corregendo gli errori in loro presenza.

* 1. **Interface Documentation Guidelines**

Per contenere gli eventuali costi di manutenzione e, allo stesso tempo, fruire di un supporto già pronto all’uso ed open source, ci avverremo di Bootstrap. In tal modo non sarà necessario impiegare tempo per le molteplici configurazioni ma sarà possibile implementare facilmente tutti gli elementi caratteristici di un front end e tra gli altri:

* + 1. **Package**

I nomi dei package sono tutti “lowerCamelCase”. Per esempio, my.exampleCode.Stragame, ma non my.examplecode.stragame o my.example\_code.stra\_game.

* + 1. **Pagine HTML**

Le pagine HTML, sia in forma statica che dinamica, devono essere conformi allo standard HTML. Inoltre, il codice HTML statico deve utilizzare l'indentazione, per facilitare la lettura, secondo le seguenti regole:

* Un’indentazione consiste in una tabulazione;
* Ogni tag deve avere un'indentazione maggiore del tag che lo contiene;
* Ogni tag di chiusura deve avere lo stesso livello di indentazione del corrispondente tag di apertura;
* I tag di commento devono seguire le stesse regole che si applicano ai tag normali.
  + 1. **Fogli di stile CSS**I fogli di stile devono essere raggruppati in opportune cartelle. Lo stile non deve essere unito con il contenuto della pagina, scritto in HTML.
    2. **Script JavaScript e JSP Indentazione**

Indentare, utilizzando un TAB, opportunamente le istruzioni.

* + 1. **Nomi**

La notazione da usare per i nomi delle variabili e dei metodi è la nota Camel Notation. I nomi dei file, delle operazioni e delle variabili devono essere evocabili.

* + 1. **Posizione dichiarazione variabili**

Posizionare le dichiarazioni all’inizio dei blocchi. Non bisogna dichiarare le variabili al loro primo uso: può confondere il programmatore inesperto e impedire la portabilità del codice dentro lo scope. L’unica eccezione a questa regola sono gli indici dei cicli for che possono essere dichiarati a più alto livello. Ad esempio, non dichiarare una variabile con lo stesso nome in un blocco interno.

* + 1. **Database SQL**

I nomi delle tabelle e dei campi devono essere: evocativi, leggibili.

* + 1. **Organizzazione dei file**

Ogni file deve essere:

* Sviluppato e diviso in base alla categoria di appartenenza, ovvero deve essere correlato ad un’unica funzionalità che persegue. Ogni pagina deve essere implementata in file separati;
* Diviso in più file se raggiunge una lunghezza tale da divenire difficile da leggere e comprendere. Organizzare in una cartella i file della libreria usate e le altre risorse scaricate necessarie per lo sviluppo del progetto.
  1. **Definizioni, Acronimi e Abbreviazioni**

Acronimi:

* RAD: Requirements Analysis Document
* SDD: System Design Document
* ODD: Object Design Document

Definizioni:

* upperCamelCase: è una convenzione di denominazione in cui un nome è formato da più parole unite insieme come una singola parola con la prima lettera di ciascuna delle parole in maiuscolo all'interno della nuova parola che forma il nome.
* lowerCamelCase: è una variante dell’upperCamelCase, in cui la prima lettera della nuova parola è minuscola,consentendo di distinguerla facilmente da un nome UpperCamelCase.
* HTML: linguaggio di programmazione utilizzato per lo sviluppo di pagine Web.
* CSS: acronimo di Cascading Style Sheets, è un linguaggio usato per definire la formattazione delle pagine Web.
* MySQL: è un DBMS relazionale, composto da varie tabelle e relazioni tra di esse.
* Bootstrap: è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il web.
* Off-The-Shelf: Servizi esterni al sistema di cui viene fatto utilizzo.
  1. **Riferimenti**
* RAD\_V.0.7.2
* SDD\_V.1.0

1. **Packages**
   1. **View**

Il package View è formato dalle pagine HTML (e relativi CSS) contententi informazioni e funzionalità per tutti gli attori del sistema. Alcuni elementi della view sono creati dinamicamente dal controller in base a specifiche necessità. Alcune di queste sono esclusive di un determinato utente, altre globali. La suddivisione è la seguente:

|  |  |
| --- | --- |
| Globali | |
| login.html | permette di effettuare il login. |
| signup.html | permette la registrazione di un nuovo utente. |
| index.html | permette di raggruppare le varie funzionalità ed è comune a tutti gli utenti. |
| viewTopics.html | permette di visualizzare la lista dei topic ordinati in base alla loro pubblicazione. |
| viewReviews.html | permette di visualizzare la lista delle recensioni ordinati in base alla loro pubblicazione. |
| viewNews.html | permette di visualizzare la lista delle news ordinati in base alla loro pubblicazione. |
| viewProfile.html | visualizzazione del profilo di un utente del forum. |
| viewVideogame.html | permette di visualizzare la lista dei prodotti videoludici presenti nel forum. |

|  |  |
| --- | --- |
| Utente | |
| newTopic.html | premette di inserire un nuovo topic. |
| newReview.html | premette di inserire una nuova recensione. |
| modifyProfile.html | permette di modificare le informazioni relative al profilo. |
| newBug.html | premette di segnalare un bug relativo ad un prodotto videoludico. |
| reportPubblication.html | permette di segnalare una pubblicazione di un utente. |

|  |  |
| --- | --- |
| Admin | |
| createNews.html | premette di creare una nuova news. |
| removePubblication.html | permette, all’admin, di rimuovere una pubblicazione che in precedenza è stata segnalata da un utente. |
| modifyProfile.html | permette di modificare le informazioni relative al profilo. |
| removeUser.html | permette, all’admin, di rimuovere un utente che ha qualche pubblicazione segnalata. |
| newVideogame.html | premette di inserire un nuovo prodotto videoludico. |

* 1. **Model**

Il package Model si occupa di fare da tramite tra l’applicazione e il database sottostante. Ogni classe contenuta all’interno del pacchetto fornisce i metodi per accedere ai dati utili all’applicazione. I moduli contenuti all’interno del package sono: User, Admin, Notification, Bug, Pubblication, News.