**Metodologia di implementazione(TUTTI)**

Tutte le funzionalità dell'applicazione sono state implementate seguendo uno sviluppo basato su più strati(MVC Pattern):

**Foundation (Tutti)**

Unico accesso alla foundation è realizzato attraverso il persistant manager, classe realizzata secondo un approccio di tipo singleton. Ogni classe della foundation ha metodi che ritornano stringhe SQL, e metodi che creano oggetti a partire da tuple che sono state recuperate dal DB. E' interessante notare l'utilizzo di funzioni che proteggono da attacchi di tipo sql injection.

Inoltre, grazie all'uso di PDO, la classe Foundation è completamente indipendente dal DBMS utilizzato.Pertanto non serve apportare ulteriori modifiche

**Entity (Tutti)**

Queste sono le classi descritte nel diagramma delle classi.Unica classe particolare è EObject, classe da cui ereditano tutte quelle classi sprovviste di un identificativo unico.Possiede metodi di validazione dei dati.

**Controller (Tutti)**

Per lo strato Controller abbiamo utilizzato un FrontController che esegue il dispatching verso altre classi controller(a seconda della richiesta).

E' interessante notare la classe CSession, la quale gestisce tutte le funzionalità di gestione delle sessioni, permettendone la creazione, distruzione e il recupero dell'utente in sessione;

**View (Tutti)**

Classi che si occupano della presentazione e di gestire Input-Output.

Unica classe particolare è VObject da cui tutte le altre classi view ereditano, classe che contiene i metodi per richiamare una pagina di errore, che inizializza un oggetto Smarty e contiene informazioni per la gestione di errori.

Infine si segnala che ogni richiesta è intercettata dallo script "index.php",il quale si occupa di (eventualmente) installare l'applicazione e invocare il FrontController.

**Fase di progettazione(TUTTI)**

Questa fase è stata portata a compimento in gruppo, e il risultato della progettazione è contenuto nei diagrammi di analisi, delle classi e schema ER.

**Fase di Implementazione**

Durante l’analisi della problematica, abbiamo notato che essa poteva essere suddivisa a sua volta in 3 sotto-problematiche ben distinte tra di loro.Perciò, nella fase di Implementazione(ovvero scrittura di codice), ognuno si è occupato di una Sezione particolare(ed eventualmente applicato delle modifiche ai diagrammi):

* Sezione Eventi - **Antonio Marottoli (n. Matricola 219204);**
* Sezione Avatar – **Alessio Perozzi (n. Matricola 252321);**
* Sezione Giochi – **Luca Del Signore (n. Matricola 252362);**

Ovviamente, le parti di codice come FrontController e FPersistentManager sono state studiate ed implementate congiuntamente.

**Installazione**

L'applicazione è stata svilupatta in PHP e provata su XAMPP, una piattaforma software che contiene strumenti quali:

* Server Apache
* Il DBMS MySQL
* PHP ( almeno Versione 7.4)

Pertanto l'applicazione dovrà richiedere un'installazione di XAMPP, in particolare la cartella contenente l'applicativo dovrà essere posizionata nella cartella *htdocs* .E’ possibile raggiungere l'applicazione attraverso l'URL localhost/playadice/ . La prima volta che vi si accede verrà visualizzata una form di installazione, in cui verranno richiesti:

* nome utente del DBMS
* password del DBMS
* nome che si vuole dare al database

Tali file verranno salvati in un file di configurazione, *config.inc.php*, che sarà utilizzato dallo strato Foundation per comunicare con il dbms. Tale procedimento, così come i meccanismi di sessione adoperati dall'applicativo, richiedono l'attivazione nel browser dei cookie.

A disposizione vi sono già tre account Admin (tutte e 3 con la stessa password:playadice)

* Badibba
* Pantaleone
* Puaison

E’ possibile cambiare la password.

Ci sono anche quattro account Utente Registrato(la cui password è quella di sopra):

* Miriairim
* Fusari
* Esperozzi
* DarkSidon

**Utenti del Sistema**

* **Visitatori(corrsispondenti a EOspite):** possono accedere al sito,visionare sezioni ”pubbliche” ma non hanno il potere di lasciare tracce(ad esempio prenotarsi ad un evento, recensire o creare un avatar).
* **Utenti Registrati(corrsispondenti a EUtente):** sono gli Utenti completi che possono usufruire interamente di tutte le funzionalità del sito. Questi possono recensire i giochi, prenotarsi agli eventi e creare Avatar.
* **Admin(corrsispondenti a EAdmin):** ha il compito di gestire il portale. Tra le sue funzioni vi sono quelle di gestione degli utenti, giochi, eventi e personaggi.

**Elenco delle funzionalità**

Di seguito è presente un elenco delle funzionalità fin'ora implementate, a cui a fianco è presente tra parentesi il nome di chi vi ha contribuito.

## Visualizzare il Catalogo dei Giochi (Luca)

### Utenti coinvolti

Tutte le tipologie di utenti

### Descrizione

Funzionalità che definisce come:

* accedere al catalogo di giochi
* fare ricerche secondo vari parametri
* visualizzare i dettagli e le recensioni dei giochi

## Gestione Catalogo e Giochi (Luca)

### Utenti coinvolti

* Admin

### Descrizione

Funzionalità che definisce come:

* aggiungere un nuovo gioco al catalogo
* rimuovere un gioco
* modificare informazioni di un gioco
* rimuovere recensioni da un gioco

## Inserire Recensioni ai Giochi(Luca)

### Utenti coinvolti

* Utente registrato
* Admin

### Descrizione

Funzionalità che definisce come:

* aggiungere la propria recensione ad un gioco
* rimuovere la propria recensione

## Gestione Profilo Utente (Luca)

### Utenti coinvolti

* Admin
* Utente Registrato

### Descrizione

Funzionalità che permette di :

* modificare Nome, Cognome ed Email
* modificare la Password
* cancellare l’Account

## Registrarsi alla piattaforma (Luca)

### Utenti coinvolti

* Ospite

### Descrizione

Funzionalità che permette di :

* Creare un nuovo Utente Registrato sul sito

## Visualizzazione e Ricerca personaggi (Alessio)

### Utenti coinvolti

* Admin
* Utente Registrato

### Descrizione

Funzionalità che permette di :

* Accedere alla lista dei propri personaggi
* Cercare tra tutti i personaggi per Username del proprietario o nome dell’Avatar
* Visualizzare i dettagli dei personaggi

## Gestione personaggi (Alessio)

### Utenti coinvolti

* Admin
* Utente Registrato

### Descrizione

Funzionalità che permette di :

* Proporre la creazione di un nuovo personaggio
* Proporre la modifica di un proprio personaggio esistente
* Proporre l’eliminazione di un proprio personaggio esistente

## Gestione proposte personaggi (Alessio)

### Utenti coinvolti

* Admin

### Descrizione

Funzionalità che permette di :

* Accettare o rifiutare proposte presentate su tutti i personaggi.

## Gestione Utenti (Alessio)

### Utenti coinvolti

* Admin

### Descrizione

Funzionalità che permette di:

* Bannare definitivamente un Utente Registrato
* Rendere Admin un Utente Registrato
* Revocare i privilegi di Admin ad un altro Admin

## Visualizzazione del Calendario (Antonio)

### Utenti coinvolti

Tutte le tipologie di Utenti

### Descrizione

Funzionalità che permette di:

* Visualizzare la lista degli Eventi;
* Permette l’ordinazione della lista per Luogo e per Data;
* Permette di accedere alla pagina del singolo evento;

## Gestione Calendario ed Evento(Antonio)

### Utenti coinvolti

* Admin

### Descrizione

Funzionalità che permette di:

* Creare un nuovo Evento dalla Lista Eventi;
* Eliminare un Evento;
* Modificare un Evento;
* Vedere le Prenotazioni ad un Evento.

## Prenotazione ad un Evento (Antonio)

### Utenti coinvolti

Tutte le tipologie di Utenti

### Descrizione

Funzionalità che permette di:

* Prenotarsi ad un Evento;
* Eliminare la propria Prenotazione.