Progetto di AMM

Matteo Zanda 47143

**Consegna:**

Pagina di descrizione del progetto con le seguenti informazioni:

1. Breve descrizione dell'applicazione (cosa fa e come mi devo aspettare che si comporti)
2. Elenco di quali requisiti del progetto avete soddisfatto. Vi ricordo che i requisiti sono i seguenti:
   1. Utilizzo di HTML e CSS
   2. Utilizzo di PHP e MySQL
   3. Utilizzo del pattern MVC
   4. Almeno due ruoli (indicare quali sono)
   5. Almeno una transazione (indicare la classe dove è implementata)
   6. Almeno una funzionalità ajax (indicare in quale script si trovi)
3. Credenziali di autenticazione e link alla homepage

Sommario

[Descrizione 2](#_Toc455233804)

[Tipologie di Utente 4](#_Toc455233805)

[Transazione 4](#_Toc455233806)

[Query function 5](#_Toc455233807)

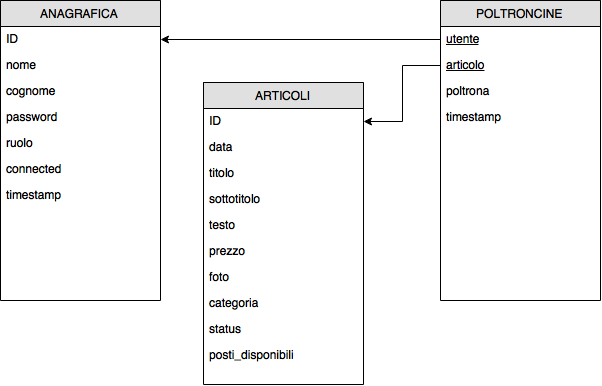
# Descrizione

L’applicazione Teatro da AMMirare è un applicativo rivolto a teatri che vogliono esporre la propria programmazione al pubblico. Le funzionalità sono rivolte alla gestione degli articoli sugli spettacoli, gestione degli utenti e di prenotazione biglietti.

Utilizzati **HTML** e **CSS** per la rappresentazione grafica della Vista,

**PHP** e **MySQL** per tutta la parte server.

Il **Database** è strutturato come segue:



E’ stato applicato il pattern MVC.

# Tipologie di Utente

Sono previste quattro tipologie di utente:

* Visitatore
* Utente Registrato -> user: user1 - pw: ciao
* Amministratore -> user: admin - pw: ciao
* SuperUser -> user: super - pw: ciao

Di seguito i privilegi consentiti ad ogni tipologia di utente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Visitatore** | **Utente Registrato** | **Amministratore** | **Super User** |
| **Visualizzazione articoli** | SI | SI | SI | SI |
| **Prenotazione**  **poltroncine** |  | SI | SI | SI |
| **Gestione articoli** |  |  | SI | SI |
| **Gestione utenti** |  |  |  | SI |

Il login è gestito con le Sessioni.

Il principale riscontro visivo della selezione di tipologia di utente si trova nella Sidebar R, che mostra la tipologia di utente, le scelte disponibili e quanti biglietti sono stati prenotati.

# Transazione

E’ stata implementata una TRANSAZIONE nella funzionalità di prenotazione dei biglietti, locata nella classe Modello. La transazione consente la memorizzazione sequenziale delle poltroncine senza errori: se durante la sequenza di query di memorizzazione il DBMS si accorge che un posto a sedere è già stato prenotato allora annulla tutta la sequenza inserita fino alla detenzione dell’errore, altrimenti conferma la memorizzazione.

# Query function

Lista di query function presenti nel Model:

* **carica\_articoli**() -> recupera tutti gli articoli
* **recupera\_articoli**() -> recupera tutti gli articoli sfruttando carica\_articoli()
* **recupera\_articolo\_per\_nome**($nomeCercato) -> recupera l’articolo cercato sfruttando carica\_articoli()
* **carica\_articoli\_pubblicati**() -> recupera tutti gli articoli che hanno status = ‘pubblicato’
* **recupera\_articoli\_pubblicati**() -> recupera gli articoli sfruttando recupera\_articoli\_pubblicati()
* **recupera\_utenti**() -> recupera tutti gli utenti
* **recupera\_utente\_per\_id**($ident) -> recupera l’utente cercato
* **recupera\_utenti\_prenotaz**() -> join tra tabella prenota e tabella anagrafica

//----------------- funzioni relative agli **ARTICOLI** -------------------------------

public function **recupera\_articoli()**

public function **recupera\_articoli\_pubblicati()**

public function **recupera\_articolo\_per\_id($idCercato)**

private function **get\_articoli()**

private function **get\_articoli\_pubblicati()**

//----------------- funzioni relative agli **UTENTI** -------------------------------

public function **recupera\_utente\_connesso()**

public function **recupera\_utenti()**

public function **recupera\_utente\_per\_id($id\_utente)** //può esserci un solo utente con l'ID cercato

public function **accedi($nome, $password)**

private function **get\_utenti()**

//----------------- funzioni relative alle **PRENOTAZIONI** -------------------------------

private function **leggi\_prenotazioni\_utente($id\_utente)**

public function **recupera\_poltroncine\_prenotate($id\_articolo)**

public function **recupera\_num\_prenotazioni\_per\_idArticolo($id\_articolo)**

private function **get\_poltroncine\_prenotate($id\_articolo)**

private function **leggi\_prenotazioni\_articolo($id\_articolo)**

public function **prenotaPoltroncine($id\_articolo, $id\_utente, $poltrone)**

private function **setPrenotaPoltroncine($id\_articolo, $id\_utente, $poltrone)**

public function **recupera\_num\_prenotazioni\_per\_idUtente($id\_utente)**