

# Diario di lavoro

Titolo progetto	Gestione ore
Luogo	Trevano-Canobbio
Data	7 novembre 2019

## Lavori svolti

Oggi ho proseguito con l'implementazione delle classi Controller, che fanno da ponte tra le classi Model e le view e viceversa.

Per essere più precisi, ho aggiunto una funzione "logout" al controller 'Home' che permetta di cancellare i dati salvati all'interno della sessione corrente, di fatto eseguendo il logout dell'utente che l'ha richiamata. Ho infatti aggiunto alla barra di navigazione un pulsante rosso che recita "LOGOUT" e richiama quest'azione.

Dopodiché ho aggiunto del codice in cima alla view 'header.php', che viene caricata come prima parte di tutte le view che si vanno a richiamare, che controlla se l'utente ha eseguito il login e, in caso contrario, lo rimanda alla homepage del sito.

```
if ($_SERVER["REQUEST_URI"] !== "/home/login") {  
    if (!(isset($_SESSION["userName"]) && isset($_SESSION["userRole"]))) {  
        header("Location: " . URL . "home");  
        die();  
    }  
}
```

E questo marcava il termine dell'implementazione del controller 'Home'. Sono poi passato al controller 'Activities', di cui ho cominciato con l'azione "index", che, come prima cosa, controlla che sia stato eseguito il login e, di conseguenza, siano stati impostati i dati all'interno della sessione, altrimenti redireziona alla homepage, poi ottiene la lista di lavori da mostrare in base al ruolo dell'utente nella piattaforma: se è un utente qualsiasi gli mostra solo i lavori a cui è stato assegnato, se invece dovesse essere un amministratore gli mostrerebbe tutti i lavori salvati nel database.

```
if (!(isset($_SESSION["userRole"]) && isset($_SESSION["userRole"]))) {  
    $this->redirect("home");  
}  
  
if ($_SESSION["userRole"] == "amministratore") {  
    $baseActivity = new Activity();  
    $activities = $baseActivity->getAllActivities();  
} else {  
    $baseAssignment = new Assignment();  
    $resource = new Resource();  
    $resource = $resource->getResourceByName($_SESSION["userName"]);  
    $activities = $baseAssignment->getActivitiesAssignedToResource($resource);  
}
```

Come è possibile notare, ho aggiunto una funzione alla classe 'Assignment' chiamata "getActivitiesAssignedToResource", che ritorna un array di oggetti di tipo 'Activity' contenente un elemento per ogni attività a cui una risorsa è stata assegnata.

Similmente, ho creato anche una funzione "getResourcesAssignedToActivity", che ritorna un array di oggetti di tipo 'Resource' contenente un elemento per ogni risorsa assegnata a un'attività.

L'unica cosa abbastanza importante da riportare è la query che viene usata per ottenere i dati dal database, visibile di seguito (in questo caso per la prima funzione, ma è molto simile anche per la seconda):

```
SELECT nome_lavoro FROM assegna WHERE nome_risorsa = :resource
```

Queste due funzioni mi hanno permesso di portare a termine la pagina corrispondente all'azione "activities/index", che ha quest'aspetto quando visualizzata dall'amministratore:

+ NUOVO LAVORO		GESTISCI RISORSE
<div>Compleanno Susan</div> <div>Data di inizio: 29.02.2004</div> <div>Data di consegna: 30.11.2005</div> <div>Ore di lavoro preventivate: 178</div> <div>Collaboratori assegnati: 1</div>	<div>Demolizione grattacielo</div> <div>Data di inizio: 03.11.2018</div> <div>Data di consegna: 15.10.2019</div> <div>Ore di lavoro preventivate: 485</div> <div>Collaboratori assegnati: 0</div>	<div>costruzione scuola</div> <div>Data di inizio: 13.05.2017</div> <div>Data di consegna: 01.11.2019</div> <div>Ore di lavoro preventivate: 789</div> <div>Collaboratori assegnati: 0</div>

Ho poi cominciato a scrivere l'azione "activities/details", ma mi sono dovuto fermare per un problema nel passaggio del nome del lavoro come parametro (vedi "Problemi riscontrati e soluzioni adottate").

#### Problemi riscontrati e soluzioni adottate

Mi sono reso conto che sotto la sezione "Programma di massima per la prossima giornata di lavoro" del rapporto della scorsa giornata di lavoro avevo lasciato il programma della lezione del 25 ottobre, senza rimuovere le attività svolte quel giorno. Purtroppo, avendo già inviato il documento al docente di riferimento, ho potuto solamente correggere il file locale.

Dopo il reindirizzamento dalla pagina di login a quella contenente la lista di lavori, i dati salvati all'interno della sessione venivano persi. Questo perché la sessione non veniva avviata, quindi ho aggiunto il seguente codice in cima al file index.php che fa partire l'applicazione:

```
if (session_status() != PHP_SESSION_ACTIVE) {
    session_start();
}
```

Questo codice controlla se la sessione è stata attivata e, in caso contrario, la avvia.

Quando ho implementato il link dalla pagina con la lista di lavori a quella dei dettagli di un lavoro ho notato che gli spazi all'interno del nome di un lavoro non venivano conservati, questo perché nella classe 'Application', facente parte del template MVC di cui ho usufruito per costruire l'applicazione, viene eseguito un filtro per controllare che l'URL sia valido prima di rimandare la richiesta alle varie classi Controller. Ho cercato invano un modo per evitare che i caratteri '+' venissero riscritti in spazi, ma non sono riuscito a riscrivere la regola all'interno del file .htaccess, quindi ho potuto solo commentare il filtro.

#### Punto della situazione rispetto alla pianificazione

Sono in anticipo rispetto alla pianificazione.

**Programma di massima per la prossima giornata di lavoro**

Portare avanti la documentazione della pianificazione e proseguire con l'implementazione delle classi Controller.