

# Diario di lavoro

Titolo progetto	Gestione ore
Luogo	Trevano-Canobbio
Data	5 novembre 2019

## Lavori svolti

Oggi ho cominciato a lavorare alle classi Controller, che fanno da ponte tra le classi Model e le view e viceversa.

Più di preciso ho completato la bozza della funzione relativa all'azione "Home/login", aggiungendo la logica di login grazie alla classe Model 'Resource' e una nuova funzione, appartenente alla superclasse 'Controller', che ho chiamato "sanitizeInput". Il codice di questa funzione è quello sottostante:

```
/**
 * Sanitizes an input string by trimming its whitespaces, filtering it and
 * encoding every character with ASCII code greater than 127 or less than 32.
 * @param string $input The input string to be sanitized.
 * @return string The sanitized input string.
 */
protected function sanitizeInput(string $input): string
{
    $input = trim($input);
    $input = filter_var($input, FILTER_SANITIZE_STRING);
    return $input;
}
```

Invece di seguito è possibile vedere le nuove aggiunte alla funzione "login", che consistono nel controllare se i dati di login sono corretti, poi salvare il nome della risorsa e il suo ruolo nella sessione:

```
//Se il tipo di richiesta è POST...
//Sanitizes and saves to a variable the username
$username = $this->sanitizeInput($_POST["nome"]);
//Sanitizes and saves to a variable the password
$password = $this->sanitizeInput($_POST["password"]);
//Create and empty resource to use its functions
$baseResource = new Resource();
//If the user exists
if ($baseResource->login($username, $password)) {
    $user = $baseResource->getResourceByName($username);
    //Add username and role to session
    $_SESSION["userName"] = $username;
    $_SESSION["userRole"] = $user->getRole();
    $this->redirect("activities");
} else {
    //Bring them back to the login form and show an error message
    $err_msg = "Il nome utente o la password inseriti non sono validi";
    require "application/views/shared/header.php";
    require "application/views/home/login.php";
    require "application/views/shared/footer.php";
}
```

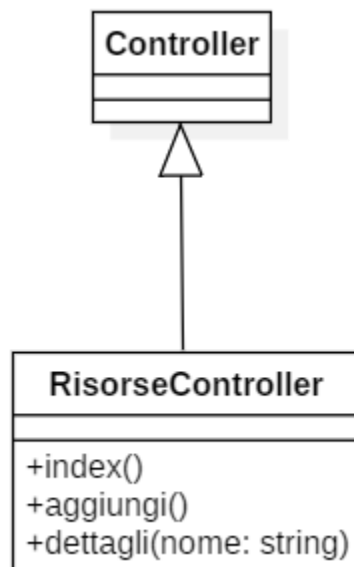
**Problemi riscontrati e soluzioni adottate**

La giornata di lavoro di oggi è iniziata scrivendo la validazione dei campi del form di login perché mi sono accorto che non l'avevo realizzata durante la fase di implementazione delle view. Semplicemente, come per le altre, mi sono limitato a utilizzare il plugin "jquery.validate" per controllare che sia stato inserito un valore sia per il nome utente che per la password e che la lunghezza dei testi inseriti sia minore a 256 caratteri. Inoltre viene usata la seguente espressione regolare per il campo 'nome', in modo da non accettare valori che potrebbero compromettere la sicurezza dell'applicazione:

```
/^[a-zAÈèìòùâëïöüâêîôûßñç&@0-9,._' ]+$/i
```

Più tardi, quando ho implementato la logica di login, ho notato che questa non funzionava perché la funzione che avevo creato nella superclasse Controller per ripulire una stringa inserita dall'utente, "sanitizeInput", ritornava sempre una stringa vuota. Questo perché fallivano le chiamate alla funzione "filter\_var" con i filtri 'FILTER\_FLAG\_STRIP\_HIGH' e 'FILTER\_FLAG\_STRIP\_LOW', quindi le ho rimosse.

Dopodiché mi sono accorto che non avevo aggiunto la superclasse 'Controller' alla progettazione delle sue sottoclassi. Di seguito la classe 'RisorseController' come esempio della modifica:

**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.