

Diario di lavoro

Titolo progetto	Gestione ore
Luogo	Trevano-Canobbio
Data	14 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 terminato l'implementazione della classe Controller 'OreLavoroController', che è nel frattempo diventata 'WorkHoursController'. Al suo interno ho implementato solo una funzione: "registra", che è diventata "register". La prima parte di questa funzione è molto simile alla funzione "assign" della classe Controller 'Assignment', infatti controlla che l'utente abbia eseguito l'accesso, altrimenti lo manda alla pagina principale, poi controlla il metodo che è stato usato per effettuare la richiesta, se è GET ottiene i dati per riempire la pagina del form di registrazione delle ore di lavoro e gliela mostra, se è POST effettua diversi controlli sulla validità dei dati inseriti e, in seguito, inserisce le ore di lavoro nel database. Di seguito la sezione che gestisce le richieste di tipo POST:

```
//Instantiate three empty objects of type Activity, Resource and Assignment respectively,
// to use their functions
$baseActivity = new Activity();
$baseResource = new Resource();
$baseAssignment = new Assignment();
//Check if a value for both the activity and the resource's names has been passed, as well as the date and
// number of hours of the work
if (isset($_POST["lavoro"]) &&
    isset($_POST["risorsa"]) &&
    isset($_POST["data"]) &&
    isset($_POST["numeroOre"])) {
    //Check if the value that was passed for the activity's name is the same as the one
    passed as parameter
    if ($_POST["lavoro"] === $activityName) {
        //Save the resource's name to a variable
        $resourceName = $this->sanitizeInput($_POST["risorsa"]);
        //Save the date on which the work took place to a variable
        $date = DateTime::createFromFormat("Y-m-d", $_POST["data"]);
        //Save the number of hours that the resource has worked to a variable
        $hoursNumber = intval($_POST["numeroOre"]);
        //Get the activity and resource with the same name as the values passed from the form
        $activity = $baseActivity->getActivityByName($activityName);
        $resource = $baseResource->getResourceByName($resourceName);
        //If the work date is invalid, show an error message, otherwise carry on with your business
        if ($date > $activity->getDeliveryDate()) {
            $err_msg =
```

```

        "Impossibile registrare le ore di lavoro: " .
        "la data di lavoro non può essere oltre quella di consegna del progetto
    .";

    } elseif ($date < $activity->getStartDate()) {
        $err_msg =
            "Impossibile registrare le ore di lavoro: " .
            "la data di lavoro non può precedere quella di inizio del progetto.";
    } else {
        //If such an activity and resource exist, add a new assignment to the datab
ase
        if (!is_null($activity) && !is_null($resource)) {
            //If the resource is actually assigned to the activity, add it to the d
atabase, otherwise
            // show an error message.
            if ($baseAssignment->isAssigned($activity, $resource)) {
                //Create a new object of type WorkHours and add it to the database
                $workHours = new WorkHours($activity, $resource, $date, $hoursNumbe
r);

                //If the addition is successful send the user to the activity's det
ails page.

                if ($workHours->addWorkHours($workHours)) {
                    $this->
>redirect("activities", "details", [urlencode($activityName)]);
                }
            } else {
                $err_msg =
                    "Impossibile registrare le ore di lavoro: " .
                    "la risorsa '$resourceName' non è assegnata al lavoro '$activit
yName'";
            }
        }
    }
}

//If no error message has been defined already, but there still was an error, show a ge
neric error message
if (!isset($err_msg)) {
    $err_msg = "Impossibile registrare le ore di lavoro.";
}

```

//If something went wrong, show the 'workHours/index' view with an error message

`$this->showIndexView($activityName, $err_msg);`

La funzione “`showIndexView`” contiene semplicemente il codice per ottenere i dati per riempire la view e per mostrarla.

Questa era l'ultima classe Controller da implementare, perciò la realizzazione di queste classi si può dire conclusa. La prossima lezione mi riservo un po' di tempo per controllare che tutti i dettagli della piattaforma siano a posto (per esempio devo adattare la barra di navigazione in modo che mostri dei collegamenti relativi alla piattaforma)

Problemi riscontrati e soluzioni adottate

Come avvenuto per la classe Controller 'Assignment', anche nella classe 'WorkHours' erano presenti due funzioni che era possibile generalizzare in una, così l'ho fatto. Ora la classe contiene solo una funzione: "register".

+registra(nomeLavoro: string)

Più tardi ho riscontrato il problema che non mi permetteva di inserire una nuova riga nella tabella 'ore_lavoro' del database. Ho poi scoperto che questo era corretto, visto che il motivo era che la risorsa non era stata assegnata al lavoro.

Poi ho notato che la gestione degli URL inesistenti avveniva nel modo seguente: se veniva inserito il nome di un Controller inesistente si veniva portati alla pagina principale, se il nome della classe Controller era reale ma non il nome dell'azione si veniva mandati alla sua funzione "index", se invece sia il nome del Controller che dell'azione corrispondevano ma non il numero di parametri necessario veniva generata un'eccezione. Ho sistemato aggiungendo una classe controller 'ErrorCodeController' che contiene la funzione "error404" che mostra una pagina di errore HTTP 404.

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 controllare che tutto ciò che è stato implementato funzioni a dovere, terminando anche i dettagli più minuziosi.

Adattare la barra di navigazione.