

Diario di lavoro

Luogo	Mendrisio
Data	20.05.2020

Lavori svolti

Durante la giornata di oggi ho fatto l'eliminazione di un corso e l'aggiunta di uno svolgimento (compresa la tabella che mostra tutti gli svolgimenti).

Per eliminare un corso utilizzo il metodo seguente:

```
public function deleteCourse($id)
{
    $selectCourse = "SELECT * FROM course WHERE id=%i";
    $courses = $this->connection->query($selectCourse, $id);
    if (count($courses) > 0) {
        $deleteCourse = "DELETE FROM course WHERE id=%i";
        $this->connection->query($deleteCourse, $id);
        echo SUCCESSFUL;
    } else {
        echo ERROR;
    }
}
```

Una volta finite la gestione dei corsi ho iniziato la gestione dell'aggiunta di uno svolgimento, a mio parere la parte più complessa del progetto. Prima di iniziare però ho effettuato una chiamata con il mio docente responsabile dove abbiamo discusso dei seguenti argomenti:

- Nell'applicazione possono venir aggiunti in modo hardcoded i docenti, essi prenderanno nella tabella user il type=2. Questo tipo di utente non interagisce con l'applicazione ma serve solamente per specificare l'utente che insegnerà al corso.
- Per insegnare un corso l'utente deve essere docente o admin.
- L'utente che vuole insegnare non può insegnare altri corsi durante lo svolgimento di un corso (non può eseguire due corsi in contemporaneo).

Il form per aggiungere uno svolgimento è questo

The screenshot shows a web application interface for managing courses. In the background, there is a table titled 'Corsi' with columns for 'TIPOLOGIA' and 'TITOLO'. A modal window titled 'Aggiungi svolgimento' is open in the foreground. The modal contains the following fields:

- Data inizio**: A date input field with the value '24.05.2020'.
- Durata Scadenza**: A date input field with the value '21.05.2020'.
- Durata corso**: A dropdown menu with the selected value '1 Giorno'.
- Insegnate**: A dropdown menu with the selected value 'gestionecorsi@yopmail.com'.
- Inserisci ora inizio e ora fine dei seguenti giorni**: A section with two input fields:
 - Inizio**: A time input field with the value '14:00'.
 - Fine**: A time input field with the value '15:00'.

At the bottom of the modal, there are two buttons: 'Aggiungi' (Add) and 'Chiudi' (Close).

Nel form va specificato la data di inizio, la durata in giorni, il docente che insegnerà lo svolgimento e i vari orari delle lezioni. La creazione di tutte le iterazioni di questo form mi ha preso tutta la giornata. Per calcolare la data di inizio minima utilizzo questo codice:

```
$.when(
  getDeadline()
).done(function (result) {
  var min = new Date();
  min.setDate(min.getDate() + Number(result) + 1);
  var dd = min.getDate();
  var mm = min.getMonth()+1;
  var yyyy = min.getFullYear();
  if(dd<10){
    dd='0'+dd;
  }
  if(mm<10){
    mm='0'+mm;
  }
  min = yyyy + '-' + mm + '-' + dd;
  var day1 = dd + '.' + mm + '.' + yyyy;
  var deadline = new Date(min);
  deadline.setDate(deadline.getDate() - Number(result));
  document.getElementById("deadline_date").innerText = getDateFormat(deadline);
  document.getElementById("in_date").value = min;
  document.getElementById("day1").innerText = day1;
  document.getElementById("in_date").setAttribute("min", min);
  document.getElementsByTagName("modal_input_add_exe").value = id;
});
```

In pratica leggo i giorni per calcolare la scadenza dell'iscrizione dal database e poi imposto in base a questo la data minima di inizio corso.

La select degli insegnati viene riempita grazie a questo metodo:

```
public function getTeachers(){
  $SelectUsers = "SELECT * FROM user WHERE type>1";
  $users = $this->connection->query($SelectUsers);
  return $users;
}
```

La parte più difficile da capire è quella di aggiungere/rimovere input time in base alla select della durata del corso:

```
function changeDay(x) {
  var count = Number(x.value);
  if (last_count < count) {
    var date = document.getElementById("in_date").value;
    var newData = new Date(date);
    for (var i = last_count + 1; i <= count; i++) {
      $('#exe_body').append('<tr id="raw_exe" + i + ">' +
        '<td>' +
        '<strong class="dL">' + getDateFormat(newData.setDate(newData.getDate()
+ 1)) + '</strong><br>' +
        'Inizio: <input id="start_time" + i + "" type="time" class="form-
control mb-1" min="06:00" max="22:00" value="14:00">' +
        'Fine: <input id="end_time" + i + "" type="time" class="form-control
mb-1" min="06:00" max="22:00" value="15:00" required>' +
        '</td>' +
```

```

        '</tr>');
    }
    last_count = count;
} else {
    for (var i = last_count; i > count; i--) {
        $('#raw_exe'+i+'').remove();
    }
    last_count = count;
}
}
}

```

Una volta preparato il form ho fatto l'aggiunta la validazione e l'aggiunta dei dati al database grazie a questo metodo:

```

public function
addExecution($start_day_lesson,$duration_day,$teacher,$times,$id_course)
{
    $data = array();
    $checkStartDayLesson = $checkDurationDay = $checkTeacher = $checkTimes =
    $checkOverlap = $checkAll = SUCCESSFUL;
    $end = "";
    $day = explode("-", $start_day_lesson);
    if (!$checkdate($day[1], $day[2], $day[0])) {
        $checkStartDayLesson = ERROR;
        $checkAll = ERROR;
    } else {
        $end = date('Y-m-d', strtotime($start_day_lesson. ' + ' . ($duration_day-1). '
days'));
        if ($this->validator->checkOverlap($start_day_lesson, $end, $teacher)) {
            $checkOverlap = ERROR;
            $checkAll = ERROR;
        }
    }
    if (!$this->validator->checkDuration($duration_day)) {
        $checkDurationDay = ERROR;
        $checkAll = ERROR;
    }
    if (!$this->validator->checkTeacher($teacher)) {
        $checkTeacher = ERROR;
        $checkAll = ERROR;
    }
    if (!$this->validator->checkTimes($times)) {
        $checkTimes = ERROR;
        $checkAll = ERROR;
    }

    if ($checkAll == SUCCESSFUL) {
        $selectCourse = "SELECT id FROM course WHERE id=%i";
        $course = $this->connection->query($selectCourse, $id_course);
        if (count($course) > 0) {
            $addExecution = "INSERT INTO execution (id_user, id_course, start, end)
VALUES (%i, %i, %s, %s)";
            $this->connection->query($addExecution, $teacher, $id_course,
$start_day_lesson, $end);
            $result = $this->connection->query("SELECT LAST_INSERT_ID() as 'id'");
            $idE = $result[0]["id"];
            $date = $start_day_lesson;
            $times = explode(" ", $times);
            for ($i = 0; $i < count($times); $i+=2){

```

```

        $addLesson = "INSERT INTO lesson (start,end,id_execution) VALUES
(%s,%s,%i)";
        $this->connection->query($addLesson, $date." ".$times[$i],$date."
".$times[$i+1], $idE);
        $date = date('Y-m-d', strtotime($date. ' + 1 days'));
    }
    }else{
        Util::fail();
    }
}
}
$data[] = array(
    SATUTS => $checkAll,
    CHECK_START_DAY => $checkStartDayLesson,
    CHECK_OVERLAP => $checkOverlap,
    CHECK_DURATION => $checkDurationDay,
    CHECK_TEACHER => $checkTeacher,
    CHECK_TIMES => $checkTimes
);
echo json_encode($data);
}

```

In questo metodo controllo la data di inizio, la durata del corso, verifico che l'utente che insegnerà il corso sia un docente o admin, verifico tutti gli intervalli di orari e infine controllo se l'utente che insegnerà il corso non sia già occupato con un altro corso, grazie a questo metodo:

```

public function checkOverlap($start,$end,$idU){
    $dStart = strtotime($start);
    $dEnd = strtotime($end);
    $selectDays = "SELECT start, end FROM execution WHERE id_user=%i";
    $days = $this->connection->query($selectDays,$idU);
    $check = 0;
    foreach ($days as $row) {
        $xstart = strtotime($row[DB_EXECUTION_START]);
        $xend = strtotime($row[DB_EXECUTION_END]);
        if (!((($dStart > $xend) || ($dEnd < $xstart))) {
            $check++;
        }
    }
    return $check != 0;
}

```

Problemi riscontrati e soluzioni adottate

Durante la lezione di oggi non ho avuto particolari problemi.

Punto della situazione rispetto alla pianificazione

Sono in linea rispetto la mia pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata di lavoro farò l'eliminazione di uno svolgimento e l'iscrizione ad un corso da parte dell'utente finale.

