

Diario di lavoro

Luogo	Balerna
Data	19.05.2020

Lavori svolti

Nella giornata di oggi, come consigliato dal perito, mi sono occupato principalmente di sviluppo quindi non ho eseguito nessun aggiornamento alla documentazione. Ho sviluppato i middleware rimanenti per ogni permesso presente nel sistema. Ad esempio il middleware InsertRequired:

```
class InsertRequired
{
    public function __invoke($request, $response)
    {
        if (!Permission::canInsert()) {
            $response->withStatus(401);
            exit;
        }
    }
}
```

Ho creato anche i middlewares chiamati SelectRequired e CreateRequired. Queste classi verranno utilizzate per determinare gli accessi ai vari percorsi. Inoltre, per la gestione dei ritardi ho dovuto creare due ulteriori classe Model una per la tabella delay ed una per la tabella student. All'interno della classe Model Students sono presenti metodi per l'inserimento e ricerca di studenti. Ad esempio il metodo che permette di ricavare uno studente dalla sua email è il seguente:

```
public static function getByEmail($email)
{
    $pdo = Database::getConnection();
    $query = "SELECT * FROM student WHERE email = :email";
    try {
        $stm = $pdo->prepare($query);
        $stm->bindParam(':email', $email);
        $stm->execute();
        return $stm->fetch(PDO::FETCH_ASSOC);
    } catch (\PDOException $e) {
        return false;
    }
}
```

Mentre nella classe di gestione per la tabella delay sono presenti tutti i metodi per gestire i ritardi, come ad esempio un metodo per ricavare tutti i ritardi in un semestre di un determinato studente in base alla sua e-mail:

```
public static function getCurrentSemesterByEmail($email)
{
    $semester = Years::getCurrentSemester();
    $start = $semester[0];
    $end = $semester[1];

    $pdo = Database::getConnection();
    $query = "SELECT id, DATE_FORMAT(date, '%d.%m.%Y') as 'date', observations,
        DATE_FORMAT(recovered, '%d.%m.%Y') as 'recovered', justified FROM
        delay WHERE email = :email AND date BETWEEN :start AND :end";

    try {
        $stm = $pdo->prepare($query);
        $stm->bindParam(':email', $email);
        $stm->bindParam(':start', $start);
        $stm->bindParam(':end', $end);
        $stm->execute();
        return $stm->fetchAll(\PDO::FETCH_ASSOC);
    } catch (\PDOException $e) {
        return false;
    }
}
```

Per entrambe le classi Modal ho creato i rispettivi Controller chiamati Delay e Student. Successivamente mi sono occupato di aggiungere come dipendenza la libreria fpdf all'interno del file composer.json:

```
"fpdf/fpdf": "dev-master"
```

In questo modo sarà possibile utilizzare questa libreria esterna per la creazione dei PDF. Per generare i PDF ho dunque creato una classe che estende FPDF la quale permetterà di generare in modo automatico i file. All'interno della classe sono presenti due metodi per creare l'header ed il footer del pdf:

```
public function Header()
{
    $this->SetFont('Arial', 'B', 12);
    $this->Cell(($this->GetPageWidth() - 20), 7, 'GESTIONE RITARDI', 'B', 0, 'T');
    $this->Ln(12);
}
```

L'header ed il footer contengono codice simile che permettono di impostare le varie pagine. All'interno del costruttore viene generato il file PDF.

Il risultato del PDF è il seguente:

GESTIONE RITARDI			
Ritardi di Bryan Beffa dal 17.01.2020 al 25.06.2020.			
Ritardi nel semestre: 3, da recuperare: 0.			
Data	Recupero	Giustificato	Osservazioni
02.02.2020	No	No	Moto
02.02.2020	No	No	Sveglia
02.02.2020	02.02.2020	No	asd

19.05.2020 Pagina 1 di 1

Figura 1 PDF generato.

Ho creato anche un ulteriore controller chiamato Home il quale si occuperà di caricare le viste del sito web che al momento non sono ancora state sviluppate. Ad esempio per caricare la pagina di gestione degli utenti il metodo è il seguente:

```
public static function users(Request $req, Response $res)
{
    $users = Users::getAll();
    return $res->render(
        __DIR__ . '/../Views/Home/users.php',
        array("users" => $users)
    );
}
```

Terminato di implementare le classi Model ed i vari Controller ho registrato i nuovi percorsi all'interno del file index.php:

```
// Percorso creazione studente.
$route->post("/student", "FilippoFinke\Controllers\Student::insert")
->before(new InsertRequired()),
// Percorso per creare un pdf dei ritardi.
$route->get("/student/{email}/pdf", "FilippoFinke\Controllers\Student::pdf")
->before(new CreateRequired()),
// Percorso per ricavare i ritardi da recuperare di uno studente.
$route->get("/student/{email}/{type}", "FilippoFinke\Controllers\Student::delays")
```

```
->before(new SelectRequired()),
// Percorso per ricavare i ritardi di uno studente.
$router->get("/student/{email}", "FilippoFinke\Controllers\Student::delays")
->before(new SelectRequired()),
// Percorso di rimozione ritardo.
$router->delete("/delay/{id}", "FilippoFinke\Controllers\Delay::delete")
->before(new InsertRequired()),
// Percorso di aggiornamento ritardo.
$router->post("/delay/{id}", "FilippoFinke\Controllers\Delay::update")
->before(new InsertRequired()),
// Percorso di aggiunta ritardo.
$router->post("/delay", "FilippoFinke\Controllers\Delay::insert")
->before(new InsertRequired())
```

Specificando già i permessi per ogni azione utilizzando le classi create in precedenza.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato durante lo sviluppo della gestione dei ritardi , degli studenti e della creazione del PDF.

Punto della situazione rispetto alla pianificazione

Essendo che ho concluso la gestione dei ritardi e la creazione dei PDF sono in anticipo di un giorno rispetto alla pianifica.

Programma di massima per la prossima giornata di lavoro

Iniziare la creazione dell'interfaccia grafica di login.