TPI 2019 - ? en php

Alexandre Benzonana
IFA-P3B
école d'informatique (CFPT-I)

4 mai 2020

Table des matières

Chapitre 1

ARL

1.1 view

addUserToLesson.php

Listing 1.1 – ./web/view/addUserToLesson.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Connection</title>
    </head>
    <body class="text-white text-center bdbg">
        <?php
            require_once 'nav.php';
        ?>
        <div class="container d-flex w-100 p-3 mx-auto flex-column align-content-center">
            <div class="">
            </div>
            <div class="col-lg-6 mx-auto mt-5">
                <?php foreach ($error as $key => $value) {
                    echo "<p class='erreur'>$value</p>";
                }?>
                <?= <<<FORMLESSON
                <form id="formulaireStudent" class="" action="" method="POST">
                    <div id="exStudentSelect">
                    $exStudentSelec
                    </div>
                    <div id="formStuden">
                        $idLessonSlect
                        $formUser
                    </div>
                    </form>
                    <button class="btn btn-outline-success" onclick="addStudentSelector()">+</button>
                    <input class="btn btn-success" name="submitSt" type="submit" onclick="validatedStudentForm()">
                    <p>$validation</p>

                FORMLESSON;
                ?>
            </div>
            <div class="col-lg-6 mx-auto mt-5">
                <?php foreach ($error as $key => $value) {
```

```
                    echo "<p class='erreur'>$value</p>";
                }?>
                <?= <<<FORMLESSON
                <form id="formulaireClass" class="" action="" method="POST
                    ">
                    <div id="exClassSelect">
                        $exClassSelec
                    </div>
                    <div id="formClass">
                        $idLessonSlect
                        $formClass
                    </div>
                </form>
                <button  class="btn btn-outline-success" onclick="addClas
                    sSelector()">+</button>
                <input class="btn btn-success" name="submitCl" type="
                    submit" onclick="validatedClassForm()">
                <p>$validation</p>
                FORMLESSON;
                ?>
            </div>
            <?php require_once "footer.html" ?>
        </div>
        <script src="js/addUserToLesson.js"></script>
    </body>
</html>
```

admin.php

Listing 1.2 – ./web/view/admin.php

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4       <meta charset="UTF-8">
5       <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
           bootstrap/4.3.1/css/bootstrap.min.css" integrity="
           sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
           iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
6       <link rel="stylesheet" href="css/style.css" type="text/css">
7       <meta name="viewport" content="width=device-width, initial-scale=1,
           shrink-to-fit=no">
8       <title>Administration</title>
9  </head>
10 <body class="bdbg text-white">
11      <?php require_once 'nav.php'; ?>
12      <div class="container d-flex w-100 h-100 p-3 mx-auto flex-column
           align-content-center">
13          <?= $display ?>
14          <div class="container">
15               <?= displayUser() ?>
16          </div>
17          <?php require_once 'footer.html'?>
18      </div>
19 </body>
20 </html>
```

calendar.php

Listing 1.3 – ./web/view/calendar.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
     <meta charset="UTF-8">
     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
     <link rel="stylesheet" href="css/style.css" type="text/css">
     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
     <title>Accueil</title>
</head>
<body class="bdbg text-white">
     <?php require_once 'nav.php'; ?>
     <div class="container mt-2">
          <div class="">
               <a href="?action=calendar&date=<?=$dateLess->format("Y-m-d")?>">
                    <button class="btn btn-outline-light mb-2"><PrÃ©cÃ©dent</button>
               </a>
               <a href="?action=calendar&date=<?=$dateAdd->format("Y-m-d")?>" class="float-right">
                    <button class="btn btn-outline-light mb-2">Suivant ></button>
               </a>
               <?= Calendrier($date) ?>
          </div>
          <?php require_once 'view/footer.html'?>
     </div>
</body>
</html>
```

changePassword.php

Listing 1.4 – ./web/view/changePassword.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Modifier mon mot de passe</title>
    </head>
    <body class="text-white text-center bdbg">
        <?php
            require_once 'nav.php';
        ?>
        <div class="container d-flex w-100 p-3 mx-auto flex-column align-content-center">
            <div class="col-lg-6 mx-auto mt-5">
            <h3>Modifier mon mot de passe</h3>
            <?= $error ?>
                <form class="" action="#" method="POST">
                    <div class="form-group">
                        <label for="Opwd">Mot de passe actuel</label>
                        <input class="form-control formInput" id="Opwd" name="oldPwd" type="password" required>
                    </div>
                    <div class="form-group">
                        <label for="pwd">Nouveau mot de passe</label>
                        <input class="form-control formInput" id="pwd" name="newPwd" type="password" required>
                    </div>
                    <div class="form-group">
                        <label for="Rpwd">RÃ©pÃ©tez le nouveau mot de passe</label>
                        <input class="form-control formInput" id="Rpwd" name="repetPwd" type="password" required>
                    </div>
                    <input class="btn btn-success" name="submit" value="Modifier" type="submit">
                </form>
            </div>
            <?php require_once "footer.html" ?>
        </div>
    </body>
</html>
```

chooseComputer.php

Listing 1.5 – ./web/view/chooseComputer.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Mon compte</title>
    </head>
    <body class="bdbg text-white">
        <?php require_once 'nav.php'; ?>
        <div class="container d-flex w-100 h-100 p-3 mx-auto flex-column align-content-center">
            <div class="container">
                <h3>Vos diffÃ©rents cours</h3>
                <form action="" method="post">
                    <?= getComputerChoises() ?>
                <input type="submit" class="btn btn-outline-primary" value="valider" name="submit" />
                </form>
            </div>
            <?php require_once 'footer.html'; ?>
        </div>
    </body>
</html>
```

createAccount.php

Listing 1.6 – ./web/view/createAccount.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" href="css/style.css" type="text/css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CrÃ©er un compte</title>
</head>
<body class="bdbg text-white">
    <?php require_once 'nav.php'; ?>
    <div class="container mt-2">
    <?= $error ?>
    <h2>CrÃ©er un compte</h2>
    <form action="#" method="post">
    <?php

        print <<<FORMULAIRE
        <div class="form-group">
            <label for="firstname">PrÃ©nom *</label>
            <input type="text" id="firstname" class="form-control" name="firstname" value="$firstname" required/>
        </div>
        <div class="form-group">
            <label for="lastname">Nom *</label>
            <input type="text" name="lastname" id="lastname" class="form-control" value="$lastname" required/>
        </div>
        <div class="form-group">
            <label for="email">Email *</label>
            <input type="email" id="email" name="email" class="form-control" value="$email" required/>
        </div>
        <div class="form-group">
            <label for="password">Mot de passe</label>
            <input type="password" id="password" name="password" class="form-control" value="" placeholder="Super par dÃ©faut" />
        </div>
        <div class="form-group">
            <label for="nfc">Tag NFC</label>
            <input type="number" name="tagNfc" id="nfc" class="form-control" value="$tag" />
        </div>
        <div class="form-group">
            <label for="class">Classe</label>
            <select id="class" class="form-control" name="class">
                <option></option>
        FORMULAIRE;

        foreach(getClasses() as $value)
        {
            $idClass = $value["idClass"];
            $className = $value['nameClass'];
            print <<<OPTION
            <option value="$idClass">$className</option>
            OPTION;
        }
    ?>
            </select>
        </div>
        <input type="submit" class="btn btn-primary" value="Ajouer" name="submit" />
    </form>
    </div>
```

```
59      <?php require_once 'footer.html';?>
60  </body>
61  </html>
```

dataView.php

Listing 1.7 – ./web/view/dataView.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=width=device-width,
            initial-scale=1.0">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
            bootstrap/4.3.1/css/bootstrap.min.css" integrity="
            sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
            iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <title>DonnÃ©es brut</title>
    </head>
    <body class="bdbg text-light">
        <?php require_once 'nav.php'; ?>

        <div class="container">
            <?php
                if(isset($error)){
                    echo '<div class="alert alert-danger mt-2" role="alert
                        ">'.$error.'</div>';
                }
            ?>
            <form class="mt-3" action='#' method='post'>
                <div class="form-row">
                    <div class="form-group col-md-6">
                        <label for="date">Date</label>
                        <input type="date" class="form-control" id="date"
                            placeholder="02-11-2019" name="date" required>
                    </div>
                    <div class="form-group">
                        <label for="inputClassNumber">Classe</label>
                        <select id="inputClassNumber" name="classId" class
                            ="form-control" required>
                            <option></option>
                            <?=
                                getClassesOptions();
                            ?>
                        </select>
                    </div>
                    <div class="form-group ml-2">
                        <label> </label>
                        <button type="submit" class="btn btn-primary for
                            m-control" value="submit" name="submit"
                            >Afficher</button>
                    </div>
                </div>
            </form>
            <table class="table table-bordered table-dark mt-3
                table-responsive-md">
                <tr>
                    <td>Code couleur</td>
                    <td class="bg-success">Normale (>=20)</td>
                    <td class="bg-primary">Un peu en dessous (<20)</td>
                    <td class="bg-warning">Non conforme (<18)</td>
                    <td class="bg-danger">Inacceptable (<=16)</td>
                </tr>
            </table>
            <?php
                if(isset($tab)){
                    echo $tab;
                }
            ?>
        </div>

        <?php require_once 'footer.html'?>
    </body>
</html>
```

editHours.php

Listing 1.8 – ./web/view/editHours.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Modifier les heures</title>
    </head>
    <body class="bdbg">
        <?php require_once 'nav.php'; ?>
        <div class="container">
            Salut
        </div>
        <?php require_once 'footer.html'; ?>
    </body>
</html>
```

footer.html

Listing 1.9 – ./web/view/footer.html

```
<footer class="mt-auto bd-footer fixed-bottom text-center w-100">
    <p>Copyright ARL Corporation &copy 2019-2020</p>
</footer>
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="
    sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n
    " crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2
    mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"
    ></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="
    sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6
    " crossorigin="anonymous"></script>
```

graphView.php

Listing 1.10 – ./web/view/graphView.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=width=device-width,
            initial-scale=1.0">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
            bootstrap/4.3.1/css/bootstrap.min.css" integrity="
            sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
            iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <title>Graphique</title>
    </head>
    <body class ="bdbg text-light">
        <?php require_once 'nav.php'; ?>

        <main class="container">
            <?php
                if($error!=null)
                    echo '<div class="alert alert-danger mt-2" role="alert
                        ">'.$error.'</div>';
            ?>
            <div class="row">
            <?php
                if($path != null){
                    echo "<img class='col-sm-12 mt-3' src='".$path."'/>";
                    echo "<a target='_blank' href='?action=downloadGraph&
                        file=".explode('/',$path)[1]."&fileType=img&
                        fileName=".$startDate."_".$endDate."' class='ml-3
                        mr-2 mt-1'><button class='btn btn-success'
                        >TÃ©lÃ©charger</button></a>";
                    echo "<a target='_blank' href='".$path."' class='mt-1'
                        ><button class='btn btn-primary'>Afficher</button>
                        </a>";
                }
            ?>
            </div>

            <form class="mt-3" action='#' method='post'>
                <div class="form-row">
                    <div class="form-group col-md-6">
                        <label for="startDate">Date de dÃ©but</label>
                        <input type="date" class="form-control" id="
                            startDate" placeholder="02-11-2019" name="
                            startDate" required>
                    </div>
                    <div class="form-group col-md-6">
                        <label for="startTime">Heure de dÃ©but <small>
                            (00:00 = dÃ©but de journÃ©e, 23:59 = fin de
                            journÃ©e)</small></label>
                        <input type="time" class="form-control" id="
                            startTime" placeholder="12:20" name="startTime"
                             value="00:00" required>
                    </div>
                    <div class="form-group col-md-6">
                        <label for="endDate">Date de fin</label>
                        <input type="date" class="form-control" id="
                            endDate" placeholder="02-11-2019" name="endDate
                            " required>
                    </div>
                    <div class="form-group col-md-6">
                        <label for="endTime">heure de fin <small>(00:00 =
                            dÃ©but de journÃ©e, 23:59 = fin de journÃ©e)</s
                            mall></label>
                        <input type="time" class="form-control" id="
                            endTime" placeholder="12:20" name="endTime"
                            value="23:59" required>
```

```
45                      </div>
46                      </div>
47                      <div class="form-group">
48                          <label for="inputClassNumber">Classe</label>
49                          <select id="inputClassNumber" name="classId" class
                                ="form-control" required>
50                           <option></option>
51                           <?=
52                               getClassesOptions();
53                           ?>
54                          </select>
55                      </div>
56                  </div>
57                  <button type="submit" class="btn btn-primary" value="
                        submit" name="submit">Afficher</button>
58              </form>
59          </main>
60          <?php require_once 'footer.html'?>
61      </body>
62  </html>
```

index.php

Listing 1.11 – ./web/view/index.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Accueil</title>
    </head>
    <body class="bdbg text-white">
        <?php require_once 'nav.php'; ?>
        <div class="container d-flex w-100 h-100 p-3 mx-auto flex-column align-content-center">
            <div class="container">
                <h1>Bienvenue sur ARL</h1>
            </div>
            <?php require_once 'footer.html';?>
        </div>
    </body>
</html>
```

interAbsCl.php

Listing 1.12 – ./web/view/interAbsCl.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
     <meta charset="UTF-8">
     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/css/bootstrap.min.css" integrity="
        sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
        iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
     <link rel="stylesheet" href="css/style.css" type="text/css">
     <meta name="viewport" content="width=device-width, initial-scale=1,
        shrink-to-fit=no">
     <title>Connection</title>
</head>
<body class="text-white text-center bdbg">
     <?php  require_once 'nav.php'; ?>
     <div class="container d-flex w-100 p-3 mx-auto flex-column
        align-content-center">
          <table>
               <tr>
                    <td></td>
                    <td></td>
               </tr>
          </table>
          <?php require_once "footer.html" ?>
     </div>
</body>
</html>
```

interAbsSt.php

Listing 1.13 – ./web/view/interAbsSt.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
     <meta charset="UTF-8">
     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
        bootstrap/4.3.1/css/bootstrap.min.css" integrity="
        sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
        iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
     <link rel="stylesheet" href="css/style.css" type="text/css">
     <meta name="viewport" content="width=device-width, initial-scale=1,
        shrink-to-fit=no">
     <title>Connection</title>
</head>
<body class="text-white text-center bdbg">
     <?php  require_once 'nav.php'; ?>
     <div class="container d-flex w-100 p-3 mx-auto flex-column
        align-content-center">
          <?= $showPresence ?>
          <?php require_once "footer.html" ?>
     </div>
</body>
</html>
```

login.php

Listing 1.14 – ./web/view/login.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Connection</title>
    </head>
    <body class="text-white text-center bdbg">
        <?php
            require_once 'nav.php';
        ?>
        <div class="container d-flex w-100 p-3 mx-auto flex-column align-content-center">
            <div class="">
                Automatic Raspberry Logger
            </div>
            <div class="col-lg-6 mx-auto mt-5">
                <?= $error ?>
                <form class="" action="#" method="POST">
                    <div class="form-group">
                        <label>Email</label>
                        <input class="form-control formInput" id="email" name="email" type="email">
                    </div>
                    <div class="form-group">
                        <label>Mot de passe</label>
                        <input class="form-control formInput" id="pwd" name="pwd" type="password">
                    </div>
                    <input class="btn btn-success" name="submit" type="submit">
                </form>
            </div>
            <?php require_once "footer.html" ?>
        </div>
    </body>
</html>
```

myAccount.php

Listing 1.15 – ./web/view/myAccount.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Mon compte</title>
    </head>
    <body class="bdbg text-white">
        <?php require_once 'nav.php'; ?>
        <div class="container d-flex w-100 h-100 p-3 mx-auto flex-column align-content-center">
            <div class="container">
                <a href="?action=changePassword"><button class="btn btn-outline-light">Modifier mon mot de passe</button></a>
                <a href="?action=chooseComputer"><button class="btn btn-outline-light">Choisir mon poste</button></a>
            </div>
            <?php require_once 'footer.html'; ?>
        </div>
    </body>
</html>
```

nav.php

Listing 1.16 – ./web/view/nav.php

```
<?php

if(count(explode('=', $_SERVER['REQUEST_URI']))>=2)
$actual = explode('=', $_SERVER['REQUEST_URI'])[1];
else
$actual = "index"

?>

<nav class="navbar navbar-dark bg-dark navbar-expand-md">
     <a class="navbar-brand" href="?action=index">ARL</a>
     <button class="navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
     </button>
     <div class="collapse navbar-collapse" id="navbarSupportedContent">
          <ul class="navbar-nav mr-auto">
               <li class="nav-item">
                    <a class="nav-link <?= $actual == 'index' ? 'active' : '' ?>" href="?action=index">Home</a>
               </li>
               <?php
               if(isset($_SESSION['log'])) {
                    echo '<li class="nav-item ' . ( $actual == 'myAccount' ? 'active' : '') . '"><a class="nav-link" href="?action=myAccount">Mon Compte</a></li>';
                    echo '<li class="nav-item ' . ( $actual == 'calendar' ? 'active' : '') . '"><a class="nav-link" href="?action=calendar">Calendrier</a></li>';

                    if ($_SESSION['role'] == 'Student') {
                    }

                    if($_SESSION['role'] == "Teacher" || $_SESSION['role'] == "Admin"){
                         echo '<li class="nav-item ' . ( $actual == 'createAccount' ? 'active' : '') . '"><a class="nav-link" href="?action=createAccount">CrÃ©er un compte</a></li>';
                    }
                    if($_SESSION['role'] == "Admin"){
                         echo '<li class="nav-item ' . ( $actual == 'schedule' ? 'active' : '') . '"><a class="nav-link" href="?action=schedule">Horaire</a></li>';
                         echo '<li class="nav-item ' . ( $actual == 'admin' ? 'active' : '') . '"><a class="nav-link" href="?action=admin">Gestion des comptes</a></li>';
                         echo '<li class="nav-item ' . ( $actual == 'scheduleStudent' ? 'active' : '') . '"><a class="nav-link" href="?action=scheduleStudent">Ajout d\'Ã©lÃ©ve a un cours</a></li>';
                    }

                    echo '<li class="nav-item dropdown">';
                    echo '<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Temperature</a>';
                    echo '<div class="dropdown-menu" aria-labelledby="navbarDropdown">';
                    echo '<a class="dropdown-item" href="?action=tempGraph">Graphique</a>';
                    echo '<a class="dropdown-item" href="?action=tempDatas">DonnÃ©es</a>';
```

```
42                      echo '<div class="dropdown-divider"></div><a class="
                            dropdown-item" href="?action=tempGlobalView">Vue d
                            \'ensemble</a></div></li>';
43
44                  }else{
45                      echo '<li class="nav-item ' . ( $actual == 'login' ? '
                            active' : '') . '"><a class="nav-link" href="?
                            action=login">Login</a></li>';
46                  }
47                  ?>
48              </ul>
49              <?= isset($_SESSION['log'])?'<a class="" href="?action=logout">
                    <button class="btn btn-outline-warning">Logout</button></a>':
                    '' ?>
50      </div>
51  </nav>
```

schedule.php

Listing 1.17 – ./web/view/schedule.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <title>Connection</title>
    </head>
    <body class="text-white text-center bdbg">
        <?php
            require_once 'nav.php';
        ?>
        <div class="container d-flex w-100 p-3 mx-auto flex-column align-content-center">
            <div class="">
            </div>
            <div class="col-lg-6 mx-auto mt-5">
                <?php foreach ($error as $key => $value) {
                    echo "<div class='alert alert-danger mt-1' role='alert'>$value</div>";
                }?>
                <?= $validation ? "<div class='alert alert-success'>$validation</div>" : ""?>
                <?= <<<FORMLESSON
                <form class="" action="" method="POST">
                    <div class="form-group">
                        <label>Nom du cour</label>
                        <input class="form-control formInput" id="namelesson" name="namelesson" type="text" value="$nameLesson" required>
                    </div>
                    <div class="form-group">
                        <label>Date de debut</label>
                        <input class="form-control formInput" id="dateDebut" name="dateDebut" type="date" min="$now" max="$dateMax" value="$dateDebut" required>
                    </div>
                    <div class="form-group">
                        <label>Date de fin</label>
                        <input class="form-control formInput" id="dateEnd" name="dateEnd" type="date" min="$now" max="$dateMax" value="$dateEnd" required>
                    </div>
                    <div class="form-group">
                        <label>Jour de la semaine de 1-5 (1->Lundi)</label>
                        <input class="form-control formInput" id="weekDay" name="weekDay" type="number" min="1" max="7" value="$weekDay" required>
                    </div>
                    <div class="form-group">
                        $timeDebutSelector
                    </div>
                    <div class="form-group">
                        $timeEndSelector
                    </div>
                    <div class="form-group">
                        $RoomSelector
                    </div>
                    <div class="form-group">
                        $TeacherSelector
                    </div>
```

```
52                      <input class="btn btn-success" name="submit" type="
                            submit">
53
54                  </form>
55                  FORMLESSON;
56                  ?>
57
58              </div>
59              <?php require_once "footer.html" ?>
60          </div>
61      </body>
62  </html>
```

temperaturesDisplay.php

Listing 1.18 – ./web/view/temperaturesDisplay.php

```php
<?php

/**
 * @return string bootstrap cards with the temperature and the bulding and class
 */
function getClassesCards(){
    $values = getLastValues();
    $cards ="";

    foreach($values as $value){
        if(count($value)>0){
            $temp = $value[0]['temperatureState'];

            $card = new Card(getBuilding($value[0]['idClass'])[0]['buildingName'], getClassName($value[0]['idClass'])[0]['classNumber'], $value[0]['dateState'], $temp, 3);
            $cards.=$card->getCard();
        }
    }
    return $cards;
}

/**
 * @param int the id of the wanted class
 * @return string bootstrap card with the temperature and the bulding and class
 */
function getClassCard($classId){
    $temp = getLastClassValue($classId)[0]['temperatureState'];
    $card = "<div class='card text-white bg-".getCardType($temp)." mt-2 col-lg-3'>";
    $card .= "<div class='card-header'>".getBuilding($classId)[0]['buildingName']." Salle ".getClassName($classId)[0]['classNumber']."</div>";
    $card .= "<div class='card-body'><h5 class='card-title'>".$temp;
    $card .= "</h5><p class='card-text'>".$value[0]['dateState']."</p></div></div>";
    return $card;
}
```

tempGlobalView.php

Listing 1.19 – ./web/view/tempGlobalView.php

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=width=device-width,
            initial-scale=1.0">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
            bootstrap/4.3.1/css/bootstrap.min.css" integrity="
            sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
            iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
        <link rel="stylesheet" href="css/style.css" type="text/css">
        <title>Accueil</title>
    </head>
    <body class="bdbg text-light">
        <?php require_once 'nav.php'; ?>
        <main class="container">
            <div class="row">
                <?php
                    echo(getClassesCards());
                ?>
            </div>
        </main>
        <?php require_once 'footer.html'?>
    </body>
</html>
```

1.2 model

addData.php

Listing 1.20 – ./web/model/addData.php

```php
<?php

$time = filter_input(INPUT_GET, "time", FILTER_SANITIZE_STRING);
$temp = filter_input(INPUT_GET, "temp", FILTER_SANITIZE_STRING);
$class = filter_input(INPUT_GET, "class", FILTER_SANITIZE_STRING);

$db = dbConnect();

$querry = dbConnect()->prepare("INSERT INTO `stateTemperatures` (`
    dateState`, `temperatureState`, `idClass`) VALUES (:dateT, :temp, :clas
    s)");
$querry -> bindParam("dateT", $time, PDO::PARAM_STR);
$querry -> bindParam("temp", $temp, PDO::PARAM_STR);
$querry -> bindParam("class", $class, PDO::PARAM_STR);
$result = $querry->execute();

/**
* Do the connection to the database
* @return PDO connection of the database
*/
function dbConnect(){
  static  $connection = null;


    $host = "hibou.lab.ecinf.ch";
  $name = "arl_temp";
  $user = "arl";
  $pass = "Super*2020";

  if($connection == null){
    try {
      $connexionString = 'mysql:host=' . $host . ';dbname=' . $name . '';
      $db = new PDO($connexionString, $user, $pass);
      $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    } catch (PDOException $e) {
      die('Erreur : ' . $e->getMessage());
    }
  }
  return $db;

}
```

addUserToLesson.php

Listing 1.21 – ./web/model/addUserToLesson.php

```php
<?php

$idUsers = filter_input(INPUT_POST, 'idUser', FILTER_DEFAULT , FILTER_REQUIRE_ARRAY);
if ($idUsers == false)
    $idUsers = [];
$idClasses = FILTER_INPUT(INPUT_POST, 'idClasse', FILTER_DEFAULT , FILTER_REQUIRE_ARRAY);
if ($idClasses == false)
    $idClasses = [];
$idLesson = FILTER_INPUT(INPUT_POST, 'idLesson', FILTER_VALIDATE_INT);

$exStudentSelec = UserSelector(0);
$exClassSelec = ClassSelector(0);

$formUser = FormUser($idUsers);
$formClass = FormClass($idClasses);
$idLessonSlect = LessonSelector($idLesson);

$error=[];
$validation = "";


if ($idUsers){
    $noError = true;

    foreach($idUsers as $key => $value)
    if (!StudentCanBeAddToLesson($value, $idLesson)){
        $noError = false;
        array_push($error, 'L\'Ã©lÃ¨ve ne peut pas Ãªtre ajoutÃ© Ã  ce cours');
    }

    if ($noError){
        $validation = "Eleve ajouter";
        foreach($idUsers as $key => $value)
        addUser_Has_Lesson($idLesson, $value);
    }
}


if ($idClasses){
    $noError = true;
    foreach($idClasses as $idC){
        $users = getClassStudents($idC);
        foreach($users as $key => $value)
        if (!StudentCanBeAddToLesson($value['idUser'], $idLesson)){
            $noError = false;
            array_push($error, 'La classe ne peut pas Ãªtre ajoutÃ© Ã  ce cours');
        }

        if ($noError){
            $validation = "Eleve ajoutÃ©";
            foreach($users as $key => $value)
            addUser_Has_Lesson($idLesson, $value['idUser']);
        }
    }
}




function FormUser($idUsers){
    $html = "";

    if (count($idUsers) > 0)
```

```
65      foreach($idUsers as $key => $value){
66          $html .= UserSelector($value);
67      }
68      else
69          $html .= UserSelector(0);
70
71      return $html;
72  }
73
74  function FormClass($idClasses){
75      $html = "";
76
77      if (count($idClasses) > 0)
78      foreach($idClasses as $key => $value){
79          $html .= ClassSelector($value);
80      }
81      else
82          $html .= ClassSelector(0);
83
84      return $html;
85  }
86
87  function UserSelector($idUserS){
88      $selector = "";
89      $rooms = getAllStudent();
90
91      $selector = '<div class="form-group"><label for="idUser[]">ElÃ©ve Ã
            ajouter :</label>';
92      $selector .= '<select name="idUser[]" class="form-control" id="idUser"
            >';
93
94      foreach ($rooms as $key => $value) {
95          $idUser = $value["idUser"];
96          $name = $value["lastname"] . " " . $value["firstname"];
97          $selector .="<option value='$idUser' ".($idUser == $idUserS ? '
                selected' : '') .">$name</option>";
98      }
99
100     $selector .= "</select></div>";
101     return $selector;
102 }
103
104 function ClassSelector($idClasseS){
105     $selector = "";
106     $rooms = getClasses();
107
108     $selector = '<div class="form-group"><label for="idClasse[]">Classe Ã
            ajouter :</label>';
109     $selector .= '<select name="idClasse[]" class="form-control" id="idCl
            asse">';
110
111     foreach ($rooms as $key => $value) {
112         $idClass= $value["idClass"];
113         $name = $value["nameClass"];
114         $selector .="<option value='$idClass' ".($idClass == $idClasseS ?
                'selected' : '') .">$name</option>";
115     }
116
117     $selector .= "</select></div>";
118     return $selector;
119 }
120
121 function LessonSelector($idLessonS){
122     $selector = "";
123     $rooms = getAllLessons();
124
125     $selector = '<div class="form-group"><label for="idLesson">Classe Ã
            ajouter :</label>';
126     $selector .= '<select name="idLesson" class="form-control" id="
            idLesson">';
127
128     foreach ($rooms as $key => $value) {
```

```
129         $idLesson = $value["idLesson"];
130         $name = $value["nameLesson"];
131         $selector .="<option value='$idLesson' ".($idLesson == $idLessonS
                ? 'selected' : '') .">$name</option>";
132     }
133 
134     $selector .= "</select></div>";
135     return $selector;
136 }
137 
138 
139 function StudentCanBeAddToLesson($idUser, $idLesson){
140     $lesson = getLessonsByIdLesson($idLesson)[0];
141 
142     $useLesson = getLessonsByIdRoomAndDateAndTime($idUser, $lesson["
            dateDebut"], $lesson["dateEnd"], $lesson["timeDebut"], $lesson["
            timeEnd"], $lesson["weekDay"]);
143 
144     return count($useLesson) == 0;
145 }
```

admin.php

Listing 1.22 – ./web/model/admin.php

```php
<?php

$done = filter_input(INPUT_GET, 'done', FILTER_SANITIZE_STRING);
$success = filter_input(INPUT_GET, 'success', FILTER_SANITIZE_STRING);
$display="";

if(!empty($done)&&!empty($success)){
     $alertType = ($success=='true'?'success':'danger');
     $messageS =  $success=='true'?'rÃ©ussit':'Ã©chouÃ©e';

     switch($done){
          case 'remove':
               $display = <<<DISPLAY
               <div class="alert alert-$alertType">
               La suppression de l'utilisateur Ã  $messageS
               </div>
               DISPLAY;
          break;
          case 'changeRole':
               $display = <<<DISPLAY
                    <div class="alert alert-$alertType">
                    Le changement de rÃ´le Ã  $messageS
                    </div>
               DISPLAY;
          break;
     }
}

function displayUser(){

     $user = getAllUsers();
     $table = '<table class="table table-dark tAdmin">';
     $table .= '<thead>';
     $table .= '<tr>';
     $table .= '<th>Nom</th>';
     $table .= '<th>PrÃ©nom</th>';
     $table .= '<th colspan="3" class="mx-auto">Roles</th>';
     $table .= '</thead>';
     $table .= '</tr>';

     foreach ($user as $key => $value) {
          $role = getUserRole($value['idUser']);

          if(!empty($role[0]['idRole'])){
               $role = $role[0]['idRole'];
          }

          $table .= "<tr>";
          $table .= '<td>'. $value['lastname'] .'</td>';
          $table .= '<td>'. $value['firstname'] .'</td>';

          //Roles isn't show dynamically
          //iduser and new role
          $table .= '<td class="tAdminBtn"><a href="?action=changeRole&id='. $value['idUser'] .'&role=Admin"><button class="';
          $table .= $role == 'Admin' ? 'btn btn-outline-success' : 'btn btn-outline-light';
          $table .= '">Administrateur</button></a></td>';

          $table .= '<td class="tAdminBtn"><a href="?action=changeRole&id='. $value['idUser'] .'&role=Teacher"><button class="';
          $table .= $role == 'Teacher' ? 'btn btn-outline-success' : 'btn btn-outline-light';
          $table .= '">Enseignant</button></td>';

          $table .= '<td class="tAdminBtn"><a href="?action=changeRole&id='. $value['idUser'] .'&role=Student"><button class="';
```

```
63            $table .= $role == 'Student' ? 'btn btn-outline-success' : 'btn
                  btn-outline-light';
64            $table .= '">Ã lÃ¨ve</button></a></td>';
65
66            $table .= '<td class="tAdminBtn"><a href="?action=removeUser&id=
                  '. $value['idUser'] .'"><button class="btn btn-outline-danger
                  ">Supprimer</button></a></td>';
67
68            $table .= '</tr>';
69        }
70        $table .= '</table>';
71        return $table;
72    }
```

calendar.php

Listing 1.23 – ./web/model/calendar.php

```php
<?php
/**
*
*/
$date = filter_input(INPUT_GET, "date", FILTER_SANITIZE_STRING);

$now = new DateTime(date("Y-m-d"));

if($date == null){
      $date = new DateTime(date("Y-m-d"));
}
else{
      $date = new DateTime($date);
}

$dateAdd = new DateTime($date->format("Y-m-d"));
$dateLess = new DateTime($date->format("Y-m-d"));

$dateAdd->add(new DateInterval('P1M'));
$dateLess->sub(new DateInterval('P1M'));

/**
*
*/
function Calendrier($date)
{
      global $now;
      $monthWord = $date->format("F");

      $date2 = new DateTime(date("d-M-Y", strtotime("first monday of ".
         $date->format("M")." ". $date->format("Y"))));
      $week = $date2->format("W");

      if($date2->format('d') != 1){
            $week -= 1;
      }
      $dayView = $date2->setISODate( 2020, $week);
      $year = $date->format("Y");

      // En-TÃªte du tableau
      $calendrierView = <<<CALENDAR
      <table class='table table-dark table-bordered' id='calendar'>
      <thead>
      <tr>
      <th id='month' colspan='7'>
      $monthWord $year
      </th>
      </tr>
      <tr>
      <th class='day'>Lundi</th>
      <th class='day'>Mardi</th>
      <th class='day'>Mercredi</th>
      <th class='day'>Jeudi</th>
      <th class='day'>Vendredi</th>
      <th class='day'>Samedi</th>
      <th class='day'>Dimanche</th>
      </tr>
      </thead>
      <tbody>
      CALENDAR;

      //Semains
      for($i = 0; $i < 6; $i++){
            $calendrierView .= "        <tr>";

            //Jours
            for($j = 0; $j < 7; $j++){
                  if($dayView->format('Y-m-d') == $now->format('Y-m-d')){
```

```
68                      $calendrierView .= "<td class='dateNow'><a class='calA
                            ' href='?action=presence&date=";
69                      $calendrierView .= $dayView->format("Y-m-d");
70                      $calendrierView .= "'><div class='calBtn'>";
71                      $calendrierView .= $dayView->format("d");
72                      $calendrierView .= "</div></a></td>";
73                  }elseif($dayView->format('m')== $date->format('m')){
74                      if ($j < 5){
75                          $calendrierView .= "<td><a class='calA' href='?
                                action=presence&date=";
76                          $calendrierView .= $dayView->format("Y-m-d");
77                          $calendrierView .= "'><div class='calBtn'>";
78                          $calendrierView .= $dayView->format("d");
79                          $calendrierView .= "</div></a></td>";
80                      }else{
81                          $calendrierView .= "<td class='bg-out'>";
82                          $calendrierView .= $dayView->format("d");
83                          $calendrierView .= "</td>";
84                      }
85                  }else{
86                      $calendrierView .= "<td class='bg-we'>";
87                      $calendrierView .= $dayView->format("d");
88                      $calendrierView .= "</td>";
89                  }
90                  $dayView->add(new DateInterval('P1D'));
91              }
92
93              $calendrierView .= "</tr>";
94          }
95          $calendrierView .= "</tbody>";
96          $calendrierView .= "</table>";
97          return $calendrierView;
98  }
```

changePassword.php

Listing 1.24 – ./web/model/changePassword.php

```php
<?php

$submited = filter_input(INPUT_POST, "submit", FILTER_SANITIZE_STRING);
$oldPwd = filter_input(INPUT_POST, "oldPwd", FILTER_SANITIZE_STRING);
$newPwd = filter_input(INPUT_POST, "newPwd", FILTER_SANITIZE_STRING);
$repetPwd = filter_input(INPUT_POST, "repetPwd", FILTER_SANITIZE_STRING);

$error = "";

if($submited == "Modifier"){
    if(checkPassword($oldPwd ,$_SESSION['id'])){

        if($oldPwd != $newPwd){
            if($newPwd == $repetPwd){
                $password = password_hash($newPwd, PASSWORD_DEFAULT);
                if(changeUserPassword($password, $_SESSION['id'])){
                    $error = "<div class='alert alert-success'>Mot de pas
                        se modifiÃ© avec succÃ¨s</div>";
                }else{
                    $error = "<div class='alert alert-warning'>Une erreure
                         est survenue</div>";
                }
            }else{
                $error = "<div class='alert alert-danger'>Les deux mots de
                    passe ne sont pas identiques</div>";
            }
        }else{
            $error = "<div class='alert alert-danger'>Le nouveau mot de p
               asse est identique Ã  l'ancien</div>";
        }
    }else{
        $error = "<div class='alert alert-danger'>Le mot de passe est inc
           orrect</div>";
    }
}
```

changeRole.php

Listing 1.25 – ./web/model/changeRole.php

```php
<?php
$idUser = FILTER_INPUT(INPUT_GET, 'id', FILTER_SANITIZE_STRING);
$idRole = FILTER_INPUT(INPUT_GET, 'role', FILTER_SANITIZE_STRING);
$success=false;
if(getRole($idUser)){
    $success = changeRole($idUser, $idRole);
}else{
    $success = setRole($idUser, $idRole);
}
$success = $success?'true':'false';

header('Location: ?action=admin&done=changeRole&success='.$success);
exit;
```

chooseComputer.php

Listing 1.26 – ./web/model/chooseComputer.php

```php
<?php
$submited = filter_input(INPUT_POST, "submit", FILTER_SANITIZE_STRING);

$dateN = date('Y-m-d');
$date = new DateTime(date('Y-m-d'));
$date->sub(new DateInterval('P7D'));
$courses = getLessonsByIdStudentAndDate($_SESSION['id'], $date->format('Y-m-d'), $dateN);

if($submited == "valider"){
    $courses = getLessonsByIdStudentAndDate($_SESSION['id'], $date->format('Y-m-d'), $dateN);
    $choosenComputers = array();
    foreach($courses as $c){
        $computer =  filter_input(INPUT_POST, 'L'.$c['idLesson'], FILTER_SANITIZE_STRING);
        $classComputer = ['idLesson'=>$c['idLesson'], 'idComputer'=>$computer];
        array_push($choosenComputers, $classComputer);
    }
    setUserComputers($_SESSION['id'], $choosenComputers);
}

function getComputerChoises(){
    global $courses;
    $form = "";
    foreach($courses as $c){
        $courseName = $c['nameLesson'];
        $idLesson = $c['idLesson'];
        $form .= <<<FORM
        <div class="form-group row">
        <label class="col-sm-3  col-form-label">$courseName</label>
        <div class="col-sm-9">
        <select name="L$idLesson" class="form-control">
        FORM;

        $computers = getComputersByRoom($c['idRoom']);
        foreach($computers as $o){
            $name = $o['nameComputer'];
            $idComputer = $o['idComputer'];
            if(computerIsSelected($_SESSION['id'], $idComputer, $idLesson)){
                $form .= <<<FORM
                <option selected value="$idComputer">$name</option>
                FORM;
            }else{
                $form .= <<<FORM
                <option value="$idComputer">$name</option>
                FORM;
            }
        }
        $form .= <<<FORM
        </select>
            </div>
            </div>
        FORM;
    }
    return $form;
}
```

crud.php

Listing 1.27 – ./web/model/crud.php

```php
<?php
DEFINE('DB_HOST', '127.0.0.1');
// DEFINE('DB_HOST', '10.5.48.40');
// DEFINE('DB_HOST', 'hibou.lab.ecinf.ch:3307');
DEFINE('DB_NAME', 'arl');
DEFINE('DB_USER', 'arl');
DEFINE('DB_PASS', 'Super*2020');

/**
* Do the connection to the database
* @return PDO connection of the database
*/
function getConnexion()
{
     static $db = null;

     if ($db === null) {
          try {
               $connexionString = 'mysql:host=' . DB_HOST . ';dbname=' .
                   DB_NAME . '';
               $db = new PDO($connexionString, DB_USER, DB_PASS);
               $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION
                   );
          } catch (PDOException $e) {
               die('Erreur : ' . $e->getMessage());
          }
     }
     return $db;
}

//Transaction functions
function startTransaction()
{
     getConnexion()->begin_transaction();
}

function rollBackTransaction()
{
     getConnexion()->rollback();
}

function commitTransaction()
{
     getConnexion()->commit();
}

/**
* Insert a user into the database
* @param string firstname of the user
* @param string lastname of the user
* @param string email of the user
* @param string password of the user
* @param string tag of the user for the scan when he's coming in class. It
    isn't mandatory
* @param int idClass is only for the student. If it's null, that mean the
    user is a teacher
* @return bool true if the insert is successful
* @return bool false if the insert isn't successful
*/
function addUser($firstname, $lastname, $email, $password, $tag, $idClass)
    {
     if(empty($tag)){
          $tag = Null;
     }
     if(empty($idClass)){
          $idClass = Null;
     }
     $connexion = getConnexion();
```

```
64      $req = $connexion->prepare("INSERT INTO users (`firstname`, `lastname
            `, `mail`, `password`, `tagNFC`, `idClass`) VALUES (:firstname, :l
            astname, :mail, :pass, :tag, :class)");
65      $req->bindParam(":firstname", $firstname, PDO::PARAM_STR);
66      $req->bindParam(":lastname", $lastname, PDO::PARAM_STR);
67      $req->bindParam(":mail", $email, PDO::PARAM_STR);
68      $req->bindParam(":pass", $password, PDO::PARAM_STR);
69      $req->bindParam(":tag", $tag, PDO::PARAM_STR);
70      $req->bindParam(":class", $idClass, PDO::PARAM_INT);
71      $req->queryString;
72      $req->execute();
73      $id = $connexion->lastInsertId();
74      if ($id > 0){
75          $req = $connexion->prepare("INSERT INTO userRole (idUser, idRole
                ) VALUES (:id, 'Student')");
76          $req->bindParam(":id", $id, PDO::PARAM_STR);
77          return $req->execute();
78      }
79      return false;
80  }
81
82  function changeUserPassword($password, $idUser){
83      $connexion = getConnexion();
84      $req = $connexion->prepare("UPDATE users SET `password`=:pass WHERE
            idUser = :idUser");
85      $req->bindParam(":pass", $password, PDO::PARAM_STR);
86      $req->bindParam(":idUser", $idUser, PDO::PARAM_STR);
87      return $req->execute();
88  }
89
90  function checkPassword($password, $idUser){
91      $connexion = getConnexion();
92      $req = $connexion->prepare('SELECT `password` from users WHERE idUser
            = :id');
93      $req->bindParam(":id", $idUser, PDO::PARAM_STR);
94      $req->execute();
95      $res = $req->fetchAll(PDO::FETCH_ASSOC);
96      if (isset($res[0])) {
97          if(password_verify($password, $res[0]['password'])){
98              return true;
99          }
100     }
101     return false;
102 }
103
104 /**
105 * Remove a user from the table userRole and users
106 * @param int id of the user
107 * @return bool true if the remove is successful
108 * @return bool false if the remove isn't successful
109 */
110 function removeUser($id){
111     $connexion = getConnexion();
112     $req = $connexion->prepare("DELETE FROM `userRole` WHERE idUser = :id
            ");
113     $req->bindParam(":id", $id, PDO::PARAM_INT);
114     if($req->execute()){
115         $req = $connexion->prepare("DELETE FROM `users` WHERE idUser = :
                id");
116         $req->bindParam(":id", $id, PDO::PARAM_INT);
117         $req->queryString;
118         return $req->execute();
119     }
120     return false;
121 }
122
123 /**
124 * Log the user with the email and the password
125 * @param string mail of the user
126 * @param string mdp of the user
127 * @return bool true if the log is successful
128 * @return bool false if the log isn't successful
```

```php
*/
function login($mail, $mdp)
{
     $connexion = getConnexion();
     $req = $connexion->prepare('SELECT `password`, idUser from users
        WHERE mail = :mail');
     $req->bindParam(":mail", $mail, PDO::PARAM_STR);
     $req->execute();
     $res = $req->fetchAll(PDO::FETCH_ASSOC);
     if (isset($res[0])) {
          if(password_verify($mdp, $res[0]['password'])){
               $_SESSION["mail"] = $mail;
               $_SESSION["id"] = $res[0]['idUser'];
               $_SESSION["role"] = getRole($res[0]['idUser']);
               return true;
          }
     }
     return false;
}

/**
* Get the role of the user with it id
* @param int idUser
* @return string the role of the user
*/
function getRole($idUser){
     $connexion = getConnexion();
     $req = $connexion->prepare('SELECT idRole from userRole WHERE idUser
        = :idUser');
     $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
     $req->execute();
     $res = $req->fetchAll(PDO::FETCH_ASSOC);
     if(isset($res[0])){
          return $res[0]['idRole'];
     }
}

/**
* Get the role of the user with it id
* @param int idUser
* @return string the role of the user
*/
function getUserRole($id){
     $conn = getConnexion();
     $req = $conn->prepare("SELECT `idRole` FROM `userRole` WHERE idUser =
         :id");
     $req->bindParam(":id", $id, PDO::PARAM_INT);
     $req->execute();
     $res = $req->fetchAll(PDO::FETCH_ASSOC);
     return $res;
}

/**
* Get the id of the user with it mail
* @param string mail of the user
* @return int the id of the user
*/
function getIdUser($mail){
     $connexion = getConnexion();
     $req = $connexion->prepare("SELECT idUser FROM users WHERE mail = :
        mail");
     $req->bindParam(":mail", $mail, PDO::PARAM_STR);
     $req->execute();
     $res = $req->fetchAll(PDO::FETCH_ASSOC);
     return $res[0]['idUser'];
}

/**
* Check if the tag the user's using is in the database
* @param string tagNFC of the user
* @return array a table assoc with the idUser
*/
```

```php
function checkTag($tag)
{
      $connexion = getConnexion();
      $req = $connexion->prepare("SELECT idUser FROM users WHERE tagNFC = :tag");
      $req->bindParam(":tag", $tag, PDO::PARAM_STR);
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get the last login of the user
* @param int idUser
* @return array a table assoc with the timeArrive, timeDepart and idPresence of the user
*/
function lastLogin($idUser)
{
      $connexion = getConnexion();
      $req = $connexion->prepare("SELECT timeArrive, timeDepart, idPresence FROM presences WHERE idUser = :idUser AND timeDepart IS NULL");
      $req->bindParam(":idUser", $idUser, PDO::PARAM_STR);
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Check the last logout of the user
* @param int idUser
* @return array a table assoc with the timeDepart and idPresence
*/
function lastLogout($idUser)
{
      $connexion = getConnexion();
      $req = $connexion->prepare("SELECT timeDepart, idPresence FROM presences WHERE idUser = :idUser AND timeDepart IS NOT NULL ORDER BY idPresence DESC LIMIT 1");
      $req->bindParam(":idUser", $idUser, PDO::PARAM_STR);
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}


/**
* Add the depart of the user
* @param int idPresence
*/
function addDepart($idPresence)
{
      $connexion = getConnexion();
      $dateNow = date('Y-m-d H:i:s');
      $req = $connexion->prepare("UPDATE presences SET timeDepart = :dateNow WHERE idPresence = :idPresence");
      $req->bindParam(":dateNow", $dateNow, PDO::PARAM_STR);
      $req->bindParam(":idPresence", $idPresence, PDO::PARAM_INT);
      $req->execute();
}

/**
* Add the arrival of the user
* @param int idPresence
*/
function addArrive($user)
{
      $connexion = getConnexion();
      $dateNow = date('Y-m-d H:i:s');
      $req = $connexion->prepare("INSERT INTO presences (timeArrive, idUser) VALUES (:arriveTime, :user)");
      $req->bindParam(":arriveTime", $dateNow, PDO::PARAM_STR);
```

```
262         $req->bindParam(":user", $user, PDO::PARAM_INT);
263         $req->execute();
264 }
265 
266 /**
267 * Get all the presences of the user
268 * @param int iduser
269 * @return array a table assoc with all the presences of the user
270 */
271 function getUserPresence($idUser)
272 {
273         $connexion = getConnexion();
274         $req = $connexion->prepare("SELECT * FROM presences WHERE idUser = :
                idUser");
275         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
276         $req->execute();
277         return $req->fetchAll(PDO::FETCH_ASSOC);
278 }
279 
280 /**
281 * Get all the presences of a class
282 * @param int idClass
283 * @return array a table assoc with all the presences of a class
284 */
285 function getClassPresence($idClass)
286 {
287         $connexion = getConnexion();
288         $req = $connexion->prepare("SELECT presences.* FROM presences, class
                es WHERE presences.idUser = classes.idUser AND classes.idClass = :
                idClass");
289         $req->bindParam(":idClass", $idClass, PDO::PARAM_INT);
290         $req->execute();
291         return $req->fetchAll(PDO::FETCH_ASSOC);
292 }
293 
294 /**
295 * Get all the presences of a day
296 * @param int date
297 * @return array a table assoc with all the presences of one day
298 */
299 function getDailyPresence($date)
300 {
301         $dateA = $date . " 00:00:00";
302         $dateD = $date . " 23:59:00";
303         $connexion = getConnexion();
304         $req = $connexion->prepare("SELECT * FROM `presences` WHERE `
                timeArrive` > :dateA AND `timeArrive` < :dateD");
305         $req->bindParam(":dateA", $dateA, PDO::PARAM_STR);
306         $req->bindParam(":dateD", $dateD, PDO::PARAM_STR);
307         $req->execute();
308         return $req->fetchAll(PDO::FETCH_ASSOC);
309 }
310 
311 /**
312 * Get all the presences of a day for the user
313 * @param int idUser
314 * @param int date
315 * @return array a table assoc with all the presences of one day
316 */
317 function getDailyUserPresence($idUser, $date)
318 {
319         $dateA = $date . " 00:00:00";
320         $dateD = $date . " 23:59:00";
321         $connexion = getConnexion();
322         $req = $connexion->prepare("SELECT * FROM `presences` WHERE `
                timeArrive` > :dateA AND `timeArrive` < :dateD AND idUser = :
                idUser");
323         $req->bindParam(":dateA", $dateA, PDO::PARAM_STR);
324         $req->bindParam(":dateD", $dateD, PDO::PARAM_STR);
325         $req->bindParam(":idUser", $idUser, PDO::PARAM_STR);
326         $req->execute();
327         return $req->fetchAll(PDO::FETCH_ASSOC);
```

```
328 }
329
330 /**
331 * Get the schedule of the user
332 * @param int idUser
333 * @return array a table assoc of all the lessons of the user
334 */
335 function getPersonSchedule($idUser)
336 {
337       $connexion = getConnexion();
338       $req = $connexion->prepare("SELECT lessons.* FROM `lessons`, `user_h
              as_lesson` WHERE (lessons.idUser = :idUser OR user_has
              _lesson.idUser = :idUser) AND user_has_Lesson.idLesson =
              lessons.idLesson");
339       $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
340       $req->execute();
341       return $req->fetchAll(PDO::FETCH_ASSOC);
342 }
343
344 /**
345 * Get the schedule of the user for one day
346 * @param int idUser
347 * @param int weekDay
348 * @param string date
349 * @return array a table assoc of all the lessons of the user
350 */
351 function getDayPersonSchedule($idUser, $weekDay, $date)
352 {
353       $connexion = getConnexion();
354       $req = $connexion->prepare("SELECT lessons.* FROM `lessons`, `user_h
              as_Lesson` WHERE lessons.weekDay = :weekDay AND dateDebut < :dateS
               AND dateEnd > :dateS AND user_has_Lesson.idLesson =
              lessons.idLesson AND (lessons.idUser = :idUser OR user_has
              _Lesson.idUser = :idUser) ");
355       $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
356       $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
357       $req->bindParam(":dateS", $date, PDO::PARAM_STR);
358       $req->execute();
359       return $req->fetchAll(PDO::FETCH_ASSOC);
360 }
361
362
363 function getComputersByRoom($idRoom)
364 {
365       $connexion = getConnexion();
366       $req = $connexion->prepare("SELECT `idComputer`, `nameComputer`, `MAC
              `, `idRoom` FROM `computers` WHERE idRoom = :idRoom");
367       $req->bindParam(":idRoom", $idRoom, PDO::PARAM_INT);
368       $req->execute();
369       return $req->fetchAll(PDO::FETCH_ASSOC);
370 }
371
372 function setUserComputers($idUser, $computers){
373       $connexion = getConnexion();
374       $req = $connexion->prepare("DELETE FROM `users_has_Computer` WHERE
              idUser = :id");
375       $req->bindParam(":id", $idUser, PDO::PARAM_INT);
376       $req->execute();
377       foreach($computers as $c){
378             $connexion = getConnexion();
379             $req = $connexion->prepare("INSERT INTO `users_has_Computer`(`
                    idUser`, `idComputer`, `idLesson`) VALUES (:idU, :idC, :idL)"
                    );
380             $req->bindParam(":idU", $idUser, PDO::PARAM_INT);
381             $req->bindParam(":idL", $c['idLesson'], PDO::PARAM_INT);
382             $req->bindParam(":idC", $c['idComputer'], PDO::PARAM_INT);
383             $req->execute();
384       }
385       return true;
386 }
387
388 function computerIsSelected($idUser, $idComputer, $idLesson){
```

```
389         $connexion = getConnexion();
390         $req = $connexion->prepare("SELECT * FROM `users_has_Computer` WHERE
                idUser = :idU AND idComputer = :idC AND idLesson = :idL");
391         $req->bindParam(":idU", $idUser, PDO::PARAM_INT);
392         $req->bindParam(":idC", $idComputer, PDO::PARAM_INT);
393         $req->bindParam(":idL", $idLesson, PDO::PARAM_INT);
394         $req->execute();
395         if(count($req->fetchAll(PDO::FETCH_ASSOC)) > 0){
396             return true;
397         }
398         return false;
399 }
400
401 /**
402 * Add a room
403 * @param string nameRoom
404 */
405 function addRoom($nameRoom)
406 {
407         $connexion = getConnexion();
408         $req = $connexion->prepare("INSERT INTO rooms (nameRoom) VALUES (:
                room)");
409         $req->bindParam(":room", $nameRoom, PDO::PARAM_STR);
410         $req->execute();
411 }
412
413 /**
414 * Return all room
415 * @return string the name of the room
416 */
417 function getAllRoom()
418 {
419         $conn = getConnexion();
420         $req = $conn->prepare("SELECT idRoom, nameRoom FROM rooms");
421         $req->execute();
422         $res = $req->fetchAll(PDO::FETCH_ASSOC);
423         return $res;
424 }
425
426 /**
427 * Search a room by the ID
428 * @return string the name of the room
429 */
430 function searchRoomById($id)
431 {
432         $conn = getConnexion();
433         $req = $conn->prepare("SELECT nameRoom FROM rooms WHERE idRoom = :id"
                );
434         $req->bindParam(":id", $id, PDO::PARAM_INT);
435         $req->execute();
436         $res = $req->fetchAll(PDO::FETCH_ASSOC);
437         return $res;
438 }
439
440 /**
441 * Get all the lessons of a room by the id of it
442 * @return array get all the information from the table lessons by the id
       of the room
443 */
444
445 function searchRoomByLesson($id)
446 {
447         $conn = getConnexion();
448         $req = $conn->prepare("SELECT nameLesson, dateDebut, dateEnd, '
                weekDay', timeDebut, timeEnd FROM lessons, rooms WHERE
                lessons.idRoom = rooms.idRoom AND lessons.idRoom = :id");
449         $req->bindParam(":id", $id, PDO::PARAM_INT);
450         $req->execute();
451         $res = $req->fetchAll(PDO::FETCH_ASSOC);
452         return $res;
453 }
454
```

```php
/**
* Get all teachers
* @return array a table assoc with the id, firstname and lastname of
    teachers
*/
function getAllTeacher()
{
      $conn = getConnexion();
      $req = $conn->prepare("SELECT users.idUser, firstname, lastname FROM
         users, userRole WHERE users.idUser = userRole.idUser and (use
         rRole.idRole = 'Teacher' OR userRole.idRole = 'Admin')");
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get all students
* @return array a table assoc with the id, firstname and lastname of
    students
*/
function getAllStudent()
{
      $conn = getConnexion();
      $req = $conn->prepare("SELECT users.idUser, firstname, lastname FROM
         users, userRole WHERE users.idUser = userRole.idUser and use
         rRole.idRole = 'Student'");
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get all users
* @return array a table assoc with the id, firstname, lastname and mail of
     users
*/
function getAllUsers(){
      $conn = getConnexion();
      $req = $conn->prepare("SELECT idUser, firstname, lastname, mail FROM
         users");
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get all roles
* @return array a table assoc with all roles
*/
function getAllRoles(){
      $conn = getConnexion();
      $req = $conn->prepare("SELECT idRole FROM roles");
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get all classes
* @return array a table assoc with all classes
*/
function getClasses(){
      $conn = getConnexion();
      $req = $conn->prepare("SELECT * FROM classes");
      $req->execute();
      $res = $req->fetchAll(PDO::FETCH_ASSOC);
      return $res;
}

/**
* Get all students in a class
```

```
519 * @param int idClass
520 * @return array a table assoc with all students
521 */
522 function getClassStudents($idClass){
523       $conn = getConnexion();
524       $req = $conn->prepare("SELECT idUser, firstname, lastname, mail,
              tagNFC FROM users WHERE idClass = :id"); //try get teacher name
525       $req->bindParam(":id", $idClass, PDO::PARAM_INT);
526       $req->execute();
527       $res = $req->fetchAll(PDO::FETCH_ASSOC);
528       return $res;
529 }
530
531 /**
532 * Get the teacher of a class
533 * @param int idClass
534 * @return array a table assoc with the id, firstname, lastname, mail and
       tagNFC of the teacher
535 */
536 function getClassteacher($idClass){
537       $conn = getConnexion();
538       $req = $conn->prepare("SELECT users.idUser, firstname, lastname, mail
              , tagNFC FROM users, classes WHERE classes.idClass = :id AND class
              es.idUser = users.idUser ");
539       $req->bindParam(":id", $idClass, PDO::PARAM_INT);
540       $req->execute();
541       $res = $req->fetchAll(PDO::FETCH_ASSOC);
542       return $res;
543 }
544
545 function getTeacherClass($teacher){
546       $conn = getConnexion();
547       $req = $conn->prepare("SELECT `idClass` FROM `classes` WHERE idUser =
              :id");
548       $req->bindParam(":id", $teacher, PDO::PARAM_INT);
549       $req->execute();
550       $res = $req->fetchAll(PDO::FETCH_ASSOC);
551       return $res;
552 }
553
554 /**
555 * Set the role of a user
556 * @param int idUser
557 * @param int idRole
558 * @return array true if the update is successful
559 */
560 function setRole($idUser, $idRole){
561       $connexion = getConnexion();
562       $req = $connexion->prepare("INSERT INTO userRole (`idUser`, `idRole`)
              VALUES (:idUser, :idRole)");
563       $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
564       $req->bindParam(":idRole", $idRole, PDO::PARAM_STR);
565       return $req->execute();
566 }
567
568 /**
569 * Change the role of the user
570 * @param int idUser
571 * @param int idRole
572 * @return bool true if success
573 */
574 function changeRole($idUser, $idRole){
575       $connexion = getConnexion();
576       $req = $connexion->prepare("UPDATE userRole SET idRole = :idRole
              WHERE idUser = :idUser");
577       $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
578       $req->bindParam(":idRole", $idRole, PDO::PARAM_STR);
579       return $req->execute();
580 }
581
582 /**
```

```
583 * Get some Lessons between dateDebut and dateEnd and between timeDebut and
        timeEnd for a user
584 * @param string nameLesson
585 * @param string dateDebut with format "YYYY-MM-DD"
586 * @param string dateEnd with format "YYYY-MM-DD"
587 * @param int weekDay
588 * @param string timeDebut with format "HH:MM"
589 * @param string timeEnd with format "HH:MM"
590 * @param int idRoom
591 * @param int idUser
592 */
593 function addLessons( $nameLesson, $dateDebut, $dateEnd, $weekDay,
        $timeDebut, $timeEnd, $idRoom, $idUser){
594     $conn = getConnexion();
595     $req = $conn->prepare("INSERT INTO lessons (nameLesson, dateDebut,
            dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser)
596     VALUE (:nameLesson, :dateDebut, :dateEnd, :weekDay, :timeDebut, :
            timeEnd, :idRoom, :idUser)");
597     $req->bindParam(":nameLesson", $nameLesson, PDO::PARAM_STR);
598     $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
599     $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
600     $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
601     $req->bindParam(":timeDebut", $timeDebut, PDO::PARAM_STR);
602     $req->bindParam(":timeEnd", $timeEnd, PDO::PARAM_STR);
603     $req->bindParam(":idRoom", $idRoom, PDO::PARAM_INT);
604     $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
605     $req->execute();
606 }
607
608 /**
609 * Get all the entry in table Lessons
610 * @return array a table assoc of the table lessons
611 */
612 function getAllLessons(){
613     $conn = getConnexion();
614     $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
            dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons"
            );
615     $req->execute();
616     $res = $req->fetchAll(PDO::FETCH_ASSOC);
617     return $res;
618 }
619
620 /**
621 * Get a Lesson by it id
622 * @param int idLesson
623 * @return array a table assoc of the table lessons
624 */
625 function getLessonsByIdLesson($idLesson){
626     $conn = getConnexion();
627     $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
            dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
            where idLesson = :idLesson");
628     $req->bindParam(":idLesson", $idLesson, PDO::PARAM_INT);
629     $req->execute();
630     $res = $req->fetchAll(PDO::FETCH_ASSOC);
631     return $res;
632 }
633
634 /**
635 * Get lessons of the user by it id
636 * @param int idUser
637 * @return array a table assoc of the table lessons
638 */
639 function getLessonsByIdUser($idUser){
640     $conn = getConnexion();
641     $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
            dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
            where idUser = :idUser");
642     $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
643     $req->execute();
644     $res = $req->fetchAll(PDO::FETCH_ASSOC);
```

```
645         return $res;
646 }
647
648 /**
649 * Get lessons of a room
650 * @param int idRoom
651 * @return array a table assoc of the table lessons
652 */
653 function getLessonsByIdRoom($idRoom){
654         $conn = getConnexion();
655         $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              where idRoom = :idRoom");
656         $req->bindParam(":idRoom", $idRoom, PDO::PARAM_INT);
657         $req->execute();
658         $res = $req->fetchAll(PDO::FETCH_ASSOC);
659         return $res;
660 }
661
662 /**
663 * Get some Lessons between dateDebut and dateEnd for a user
664 * @param string date au format "YYYY-MM-DD"
665 * @return array
666 */
667 function getLessonsByDate($date){
668         $conn = getConnexion();
669         $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              where :date BETWEEN dateDebut and dateEnd");
670         $req->bindParam(":date", $date, PDO::PARAM_STR);
671         $req->execute();
672         $res = $req->fetchAll(PDO::FETCH_ASSOC);
673         return $res;
674 }
675
676 /**
677 * Get some Lessons between dateDebut and dateEnd for a teacher
678 * @param int idUser
679 * @param string date au format "YYYY-MM-DD"
680 * @return array
681 */
682 function getLessonsByIdTeacherAndDate($idUser, $date){
683         $conn = getConnexion();
684         $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              where idUser = :idUser AND :date BETWEEN dateDebut AND dateEnd");
685         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
686         $req->bindParam(":date", $date, PDO::PARAM_STR);
687         $req->execute();
688         $res = $req->fetchAll(PDO::FETCH_ASSOC);
689         return $res;
690 }
691
692 /**
693 * Get some Lessons between dateDebut and dateEnd for a teacher
694 * @param int idUser
695 * @param string date au format "YYYY-MM-DD"
696 * @return array
697 */
698 function getLessonsByIdStudentAndDate($idUser, $dateD, $dateE){
699         $conn = getConnexion();
700         $req = $conn->prepare("SELECT lessons.idLesson, nameLesson, dateDebut
              , dateEnd, weekDay, timeDebut, timeEnd, idRoom FROM lessons, use
              r_has_Lesson WHERE user_has_Lesson.idLesson = lessons.idLesson AND
               user_has_Lesson.idUser = :idUser AND dateDebut <= :dateD AND
              dateEnd >= :dateE ");
701         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
702         $req->bindParam(":dateD", $dateD, PDO::PARAM_STR);
703         $req->bindParam(":dateE", $dateE, PDO::PARAM_STR);
704         $req->execute();
705         $res = $req->fetchAll(PDO::FETCH_ASSOC);
706         return $res;
```

```
707 }
708
709 /**
710 * Get some Lessons between dateDebut and dateEnd for a room
711 * @param int idRoom
712 * @param string date au format "YYYY-MM-DD"
713 * @return array
714 */
715 function getLessonsByIdRoomAndDate($idRoom, $date){
716       $conn = getConnexion();
717       $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              WHERE idRoom = :idRoom AND :date BETWEEN dateDebut AND dateEnd");
718       $req->bindParam(":idRoom", $idRoom, PDO::PARAM_INT);
719       $req->bindParam(":date", $date, PDO::PARAM_STR);
720       $req->execute();
721       $res = $req->fetchAll(PDO::FETCH_ASSOC);
722       return $res;
723 }
724
725 /**
726 * Get some Lessons between dateDebut and dateEnd and between timeDebut and
        timeEnd
727 * @param string date au format "YYYY-MM-DD"
728 * @param string time au format "HH:MM"
729 * @return array
730 */
731 function getLessonsByDateAndTime($date, $time){
732       $conn = getConnexion();
733       $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              WHERE :date BETWEEN dateDebut and dateEnd AND :time BETWEEN
              timeDebut AND timeEnd");
734       $req->bindParam(":date", $date, PDO::PARAM_STR);
735       $req->bindParam(":time", $time, PDO::PARAM_STR);
736       $req->execute();
737       $res = $req->fetchAll(PDO::FETCH_ASSOC);
738       return $res;
739 }
740
741 /**
742 * Get some Lessons between dateDebut and dateEnd and between timeDebut and
        timeEnd for a user
743 * @param int idUser
744 * @param string dateDebut with format "YYYY-MM-DD"
745 * @param string dateEnd with format "YYYY-MM-DD"
746 * @param int weekDay
747 * @param string timeDebut with format "HH:MM"
748 * @param string timeEnd with format "HH:MM"
749 * @return array
750 */
751 function getLessonsByIdUserAndDateAndTime($idUser, $dateDebut, $dateEnd,
       $weekDay, $timeDebut, $timeEnd){
752       $conn = getConnexion();
753       $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
              dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
              WHERE idUser = :idUser AND ((:dateDebut BETWEEN dateDebut and
              dateEnd OR :dateEnd BETWEEN dateDebut and dateEnd) OR (dateDebut
              BETWEEN :dateDebut and :dateEnd OR dateEnd BETWEEN :dateDebut and
              :dateEnd) ) AND ((:timeDebut BETWEEN timeDebut and timeEnd OR :
              timeEnd BETWEEN timeDebut and timeEnd) OR (timeDebut BETWEEN :
              timeDebut and :timeEnd OR timeEnd BETWEEN :timeDebut and :timeEnd)
              ) and :weekDay = weekday");
754
755       $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
756       $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
757       $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
758       $req->bindParam(":timeDebut", $timeDebut, PDO::PARAM_STR);
759       $req->bindParam(":timeEnd", $timeEnd, PDO::PARAM_STR);
760       $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
761
762       $req->execute();
```

```
763         $res = $req->fetchAll(PDO::FETCH_ASSOC);
764         return $res;
765 
766 }
767 
768 /**
769 * Get some Lessons between dateDebut and dateEnd and between timeDebut and
        timeEnd for a room
770 * @param int idRoom
771 * @param string dateDebut with format "YYYY-MM-DD"
772 * @param string dateEnd with format "YYYY-MM-DD"
773 * @param int weekDay
774 * @param string timeDebut with format "HH:MM"
775 * @param string timeEnd with format "HH:MM"
776 * @return array
777 */
778 function getLessonsByIdRoomAndDateAndTime($idRoom, $dateDebut, $dateEnd,
        $timeDebut, $timeEnd, $weekDay){
779         $conn = getConnexion();
780         $req = $conn->prepare("SELECT idLesson, nameLesson, dateDebut,
            dateEnd, weekDay, timeDebut, timeEnd, idRoom, idUser FROM lessons
            where idRoom = :idRoom AND ((:dateDebut BETWEEN dateDebut and
            dateEnd OR :dateEnd BETWEEN dateDebut and dateEnd) OR (dateDebut
            BETWEEN :dateDebut and :dateEnd OR dateEnd BETWEEN :dateDebut and
            :dateEnd) ) AND ((:timeDebut BETWEEN timeDebut and timeEnd OR :
            timeEnd BETWEEN timeDebut and timeEnd) OR (timeDebut BETWEEN :
            timeDebut and :timeEnd OR timeEnd BETWEEN :timeDebut and :timeEnd)
            ) AND :weekDay = weekday;");
781 
782         $req->bindParam(":idRoom", $idRoom, PDO::PARAM_INT);
783         $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
784         $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
785         $req->bindParam(":timeDebut", $timeDebut, PDO::PARAM_STR);
786         $req->bindParam(":timeEnd", $timeEnd, PDO::PARAM_STR);
787         $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
788         $req->execute();
789         $res = $req->fetchAll(PDO::FETCH_ASSOC);
790         return $res;
791 }
792 
793 // Presence
794 
795 /**
796 * Get all presence for a teacher by the Student id
797 * @param int idStudent
798 * @param string dateDebut with format "YYYY-MM-DD"
799 * @param string dateEnd with format "YYYY-MM-DD"
800 * @param int weekDay
801 * @param string timeDebut with format "HH:MM"
802 * @param string timeEnd with format "HH:MM"
803 * @return array
804 */
805 function getTeacherPresenceByStudentIdDayWeekDayTime($idStudent,
        $dateDebut, $dateEnd, $weekDay, $timeDebut, $timeEnd){
806         $conn = getConnexion();
807         $req = $conn->prepare("SELECT presences.timeArrive as timeArrive,
            presences.timeDepart as timeDepart From users, lessons, presences
            WHERE ( presences.timeArrive  BETWEEN :dateDebut and :dateEnd OR
            presences.timeDepart BETWEEN :dateDebut and :dateEnd) and use
            rs.idUser = presences.idUser and lessons.idUser = users.idUser and
             lessons.idLesson = (SELECT lessons.idLesson from users, user_has
            _Lesson, lessons WHERE users.idUser = user_has_Lesson.idUser and
            users.idUser = :idStudent and user_has_Lesson.idLesson =
            lessons.idLesson and :dateDebut BETWEEN lessons.dateDebut AND
            lessons.dateEnd and lessons.weekDay = :weekDay and
            lessons.timeDebut = :timeDebut and lessons.timeEnd = :timeEnd);");
808 
809         $req->bindParam(":idStudent", $idStudent, PDO::PARAM_INT);
810         $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
811         $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
812         $req->bindParam(":timeDebut", $timeDebut, PDO::PARAM_STR);
813         $req->bindParam(":timeEnd", $timeEnd, PDO::PARAM_STR);
```

```
814         $req->bindParam(":weekday", $weekDay, PDO::PARAM_INT);
815
816         $req->execute();
817         $res = $req->fetchAll(PDO::FETCH_ASSOC);
818         return $res;
819 }
820
821
822 /**
823 * Get all presence for a student between tow DateTime
824 * @param int idStudent
825 * @param string dateDebut with format "YYYY-MM-DD"
826 * @param string dateEnd with format "YYYY-MM-DD"
827 * @return array
828 */
829 function getStudentPresenceByStudentIdDateTime($idStudent, $dateDebut,
        $dateEnd){
830         $conn = getConnexion();
831         $req = $conn->prepare("SELECT  presences.timeArrive as timeArrive,
            presences.timeDepart as timeDepart From users, presences WHERE use
            rs.idUser = presences.idUser and users.idUser = :idStudent and(
            presences.timeArrive  BETWEEN :dateDebut and :dateEnd OR
            presences.timeDepart BETWEEN :dateDebut and :dateEnd)");
832
833         $req->bindParam(":idStudent", $idStudent, PDO::PARAM_INT);
834         $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
835         $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
836
837         $req->execute();
838         $res = $req->fetchAll(PDO::FETCH_ASSOC);
839         return $res;
840 }
841
842 /**
843 * Get all presence for a student between tow DateTime
844 * @param int idStudent
845 * @param string dateDebut with format "YYYY-MM-DD"
846 * @param string dateEnd with format "YYYY-MM-DD"
847 * @param int weekDay
848 * @param string timeDebut with format "HH:MM"
849 * @param string timeEnd with format "HH:MM"
850 * @return array
851 */
852 function getidLessonByidStudent($idStudent, $dateDebut, $dateEnd,
        $timeDebut, $timeEnd, $weekDay){
853         $conn = getConnexion();
854         $req = $conn->prepare("SELECT lessons.idLesson From lessons, user_has
            _Lesson WHERE user_has_Lesson.idUser = :idStudent and user_has
            _Lesson.idLesson = lessons.idLesson AND ((:dateDebut BETWEEN
            dateDebut and dateEnd OR :dateEnd BETWEEN dateDebut and dateEnd)
            OR (dateDebut BETWEEN :dateDebut and :dateEnd OR dateEnd BETWEEN :
            dateDebut and :dateEnd) ) AND ((:timeDebut BETWEEN timeDebut and
            timeEnd OR :timeEnd BETWEEN timeDebut and timeEnd) OR (timeDebut
            BETWEEN :timeDebut and :timeEnd OR timeEnd BETWEEN :timeDebut and
            :timeEnd)) and lessons.weekDay = :weekDay");
855
856         $req->bindParam(":idStudent", $idStudent, PDO::PARAM_INT);
857         $req->bindParam(":dateDebut", $dateDebut, PDO::PARAM_STR);
858         $req->bindParam(":dateEnd", $dateEnd, PDO::PARAM_STR);
859         $req->bindParam(":timeDebut", $timeDebut, PDO::PARAM_STR);
860         $req->bindParam(":timeEnd", $timeEnd, PDO::PARAM_STR);
861         $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
862
863         $req->execute();
864         $res = $req->fetchAll(PDO::FETCH_ASSOC);
865         return $res;
866
867 }
868
869 function addUser_Has_Lesson($idLesson, $idUser){
870         $conn = getConnexion();
871         $req = $conn->prepare("INSERT INTO user_has_Lesson (idLesson, idUser)
```

```
872         VALUE (:idLesson, :idUser)");
873         $req->bindParam(":idLesson", $idLesson, PDO::PARAM_INT);
874         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
875         $req->execute();
876 }
877
878 function didUserHaveClasses($idUser){
879         $timeN = date('H:i');
880         $weekDay = date('N');
881         $dateN = date('Y-m-d');
882         $conn = getConnexion();
883         $req = $conn->prepare("SELECT * FROM lessons, user_has_Lesson WHERE
                user_has_Lesson.idUser = :idUser AND `weekDay` = :weekDay AND `
                timeDebut` <= :timeN AND `timeEnd` >= :timeN AND `dateDebut` <= :
                dateN AND `dateEnd` >= :dateN");
884         $req->bindParam(":idUser", $idUser, PDO::PARAM_INT);
885         $req->bindParam(":weekDay", $weekDay, PDO::PARAM_INT);
886         $req->bindParam(":timeN", $timeN, PDO::PARAM_STR);
887         $req->bindParam(":dateN", $dateN, PDO::PARAM_STR);
888         $req->execute();
889         return $req->fetchAll(PDO::FETCH_ASSOC);
890 }
```

dataBase.php

Listing 1.28 – ./web/model/dataBase.php

```php
<?php
/**
 * @author Alexandre Benzonana
 * @version 1.2
 */

/**
 * Connect to database
 * @return connection to the DB
 */
function dbConnect(){
  static  $connection = null;

    $host = "127.0.0.1";
    // $host = "hibou.lab.ecinf.ch:3307";
  $name = "arl_temp";
  $user = "arl";
  $pass = "Super*2020";

  if($connection == null){
    try {
            $connexionString = 'mysql:host=' . $host . ';dbname=' . $name
                . ';charset=utf8';
            $db = new PDO($connexionString, $user, $pass);
            $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        } catch (PDOException $e) {
            die('Erreur : ' . $e->getMessage());
        }
    }
    return $db;

}

/**
 * Get all datas that have been stored
 * @return array all the temperature datas stored in the DB
 */
function getAllDatas(){
    static $querry = null;

    $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`");
    $result = $querry->execute();
    return $result->fetchAll(PDO::FETCH_ASSOC);
}

/**
 * Get datas between start and end params of the actual day
 * @param string $startTime time formated string for the start of the data
     selection (hh:mm)
 * @param string $endTime time formated string for the end of the data
     selection (hh:mm)
 * @return array all the datas of the day an hours that match the params
 */
function getDailyDatas(int $classId, string $startTime="00:00", string
    $endTime="23:59:59"){
    static $querry = null;

    $day = date("Y-m-d");
    $startTime=$day." ".$startTime;
    $endTime=$day." ".$endTime;

    $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`
       WHERE `dateState` BETWEEN :start AND :end AND idClass = :classId");
    $querry -> bindParam("start", $startTime, PDO::PARAM_STR);
    $querry -> bindParam("end", $endTime, PDO::PARAM_STR);
    $querry -> bindParam("classId", $classId, PDO::PARAM_INT);
    $querry->execute();
    return $querry->fetchAll(PDO::FETCH_ASSOC);
```

```
64
65 }
66
67 /**
68  * Get all datas contained between start and end params
69  * @param string $startDate dateTime format for the start of the data
       selection (yyyy-mm-dd hh:mm)
70  * @param string $endDate dateTime format for the end of the data
       selection (yyyy-mm-dd hh:mm)
71  * @param int classId
72  * @return array all the datas stored in the DB that match the params
73  */
74 function getClassDatas($startDate, $endDate, $classId){
75     static $querry = null;
76
77     //REGEX for the date with time
78     $re1='((?:(?:[1]{1}\\d{1}\\d{1}\\d{1})|(?:[2]{1}\\d{3}))[-:
          \\/.](?:[0]?[1-9]|[1][012])[-:\\/.](?:(?:[0-2]?\\d{1})
          |(?:[3][01]{1})))(?![\\d])';
79     $re2='(\\s+)';
80     $re3='((?:(?:[0-1][0-9])|(?:[2][0-3])|(?:[0-9])):(?:[0-5][0-9]))';
81
82     if (!(preg_match_all ("/".$re1.$re2.$re3."/is", $startDate) && (
          preg_match_all ("/".$re1.$re2.$re3."/is", $endDate))))
83     {
84         return "Invalid datetime format --> expected 'YYYY-MM-DD HH:MM'";
85     }
86
87     $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`
          WHERE `dateState` BETWEEN :start AND :end AND idClass = :classId");
88     $querry -> bindParam("start", $startDate, PDO::PARAM_STR);
89     $querry -> bindParam("end", $endDate, PDO::PARAM_STR);
90     $querry -> bindParam("classId", $classId, PDO::PARAM_INT);
91     $querry->execute();
92     return $querry->fetchAll(PDO::FETCH_ASSOC);
93 }
94
95 /**
96  * Get all datas for the specified class
97  * @param int classId
98  * @return array all the datas stored in the DB that match the classId
99  */
100 function getAllClassDatas($classId){
101     static $querry = null;
102
103     $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`
          WHERE `idClass` = :classId");
104     $querry -> bindParam("classId", $classId, PDO::PARAM_INT);
105     $querry->execute();
106     return $querry->fetchAll(PDO::FETCH_ASSOC);
107 }
108
109 /**
110 * Get all classes
111 * @return array of all classes with id and building id
112 */
113 function getTempClasses(){
114     static $querry = null;
115
116     $querry = dbConnect()->prepare("SELECT * FROM `classes`");
117     $querry->execute();
118     return $querry->fetchAll(PDO::FETCH_ASSOC);
119 }
120
121 /**
122  * Get all buildings
123  * @return array of all buildings with their id
124  */
125 function getBuildings(){
126     static $querry = null;
127
128     $querry = dbConnect()->prepare("SELECT * FROM `buildings`");
```

```
129     $querry->execute();
130     return $querry->fetchAll(PDO::FETCH_ASSOC);
131 }
132 
133 /**
134  * Get all classes in a building
135  * @param int id of the building
136  * @return array of all classes in the specified building
137  */
138 function getBuildingClasses($buildingId){
139     static $querry = null;
140 
141     $querry = dbConnect()->prepare("SELECT * FROM `classes` WHERE
            idBuilding = :idBuilding");
142     $querry -> bindParam("classId", $buildingId, PDO::PARAM_INT);
143     $querry->execute();
144     return $querry->fetchAll(PDO::FETCH_ASSOC);
145 }
146 
147 /**
148  * Get the last measured value foreach class
149  * @return array of temperatures for all classes
150  */
151 function getLastValues(){
152     static $querry = null;
153     $values = array();
154     $classes = getClasses();
155 
156     foreach($classes as $value){
157         $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`
                WHERE idClass = :idClass ORDER BY `idStateTemperature` DESC
                LIMIT 1");
158         $querry -> bindParam("idClass", $value['idClass'], PDO::PARAM_INT)
                ;
159         $querry->execute();
160         array_push($values, $querry->fetchAll(PDO::FETCH_ASSOC));
161     }
162     return $values;
163 }
164 
165 /**
166  * Get the last measured value of the given class
167  * @param int of the wanted class
168  * @return array with temp, class, building and date
169  */
170 function getLastClassValue($classId){
171     static $querry = null;
172 
173     $querry = dbConnect()->prepare("SELECT * FROM `stateTemperatures`
            WHERE idClass = :idClass ORDER BY `idStateTemperature` DESC LIMIT 1
            ");
174     $querry -> bindParam("idClass", $classId, PDO::PARAM_INT);
175     $querry->execute();
176     return $querry->fetchAll(PDO::FETCH_ASSOC);
177 }
178 
179 /**
180  * Get class number with given id
181  * @param int of wanted class
182  * @return array with class number
183  */
184 function getClassName(int $classId){
185     static $querry = null;
186 
187     $querry = dbConnect()->prepare("SELECT `classNumber` FROM `classes`
            WHERE idClass = :idClass");
188     $querry -> bindParam("idClass", $classId, PDO::PARAM_INT);
189     $querry->execute();
190     return $querry->fetchAll(PDO::FETCH_ASSOC);
191 }
192 
193 /**
```

```
194  *
195  */
196 function getBuilding(int $classId){
197     static $querry = null;
198     $idBuilding = getBuildingId($classId)[0]['idBuilding'];
199 
200     $querry = dbConnect()->prepare("SELECT `buildingName` FROM `buildings`
            WHERE idBuilding = :idBuilding");
201     $querry -> bindParam("idBuilding", $idBuilding, PDO::PARAM_INT);
202     $querry->execute();
203     return $querry->fetchAll(PDO::FETCH_ASSOC);
204 }
205 
206 /**
207  *
208  */
209 function getBuildingId(int $classId){
210     static $querry = null;
211 
212     $querry = dbConnect()->prepare("SELECT `idBuilding` FROM `classes`
           WHERE idClass = :idClass");
213     $querry -> bindParam("idClass", $classId, PDO::PARAM_INT);
214     $querry->execute();
215     return $querry->fetchAll(PDO::FETCH_ASSOC);
216 }
217 
218 /**
219  *
220  */
221 function storeGraph(string $startDate, string $endDate, string $graphPath,
        int $classId){
222     static $querry = null;
223 
224     $querry = dbConnect()->prepare("INSERT INTO `generatedgraphs`(`
           startDate`, `endDate`, `graphPath`, `classId`) VALUES (:startDate,:
           endDate,:graphPath,:idClass)");
225     $querry -> bindParam("startDate", $startDate, PDO::PARAM_STR);
226     $querry -> bindParam("endDate", $endDate, PDO::PARAM_STR);
227     $querry -> bindParam("graphPath", $graphPath, PDO::PARAM_STR);
228     $querry -> bindParam("idClass", $classId, PDO::PARAM_INT);
229     $ok = $querry->execute();
230     return $ok;
231 }
232 
233 /**
234  *
235  */
236 function graphExist(string $startDate, string $endDate, int $classId){
237     static $querry = null;
238 
239     $querry = dbConnect()->prepare("SELECT * FROM `generatedgraphs` WHERE
           `startDate` = :startDate AND `endDate` = :endDate AND `classId` = :
           classId");
240     $querry -> bindParam("startDate", $startDate, PDO::PARAM_STR);
241     $querry -> bindParam("endDate", $endDate, PDO::PARAM_STR);
242     $querry -> bindParam("classId", $classId, PDO::PARAM_INT);
243     $querry->execute();
244     return $querry->fetchAll(PDO::FETCH_ASSOC);
245 }
```

dataView.php

Listing 1.29 – ./web/model/dataView.php

```php
<?php

$date = filter_input(INPUT_POST, 'date', FILTER_SANITIZE_STRING);
$classId = filter_input(INPUT_POST, 'classId', FILTER_SANITIZE_NUMBER_INT);
$tab = null;

if(count(explode('-', $date))!=3){
    $date = date('Y-m-d');
}


if($classId == null){
    $classId = 1;
}

if($date!=null && $classId != null){
    try{
        $values = getClassDatas($date.' 00:00', $date.' 23:59', $classId);
        if(count($values) > 0){
            $tab = getDataTab($date, $values);
        }else{
            $error = 'Aucune donnÃ©e n\'a Ã©tÃ© trouvÃ©e avec les paramÃ¨tres indiquÃ©s';
        }
    }catch(Exception  $e){

    }
}

function getDataTab($date, $values){
    $returnTab = '<table class="table table-bordered table-dark mt-3 table-responsive-md"> <thead> <tr><th scope="col" class="text-center h3 font-weight-bold" colspan="7">'.$date.'</th></tr><tr><th scope="col">Heure / minute</th>';

    for($i=0; $i<6; $i++){
        $returnTab.='<th scope="col">'.$i.'0</th>';
    }
    $returnTab."</tr></thead><tbody>";
    foreach($values as $value){
        $date = explode(" ", $value['dateState']);
        $hour = explode(":", $date[1]);

        if($hour[1]=="00"){
            $returnTab."<tr>";
                    $returnTab.='<td scope="row">'.$hour[0].'</th>';
            }
            $color = "bg-success";
            if($value['temperatureState'] <= 16)
                $color = "bg-danger";
            elseif($value['temperatureState'] <18)
            $color = "bg-warning";
            elseif($value['temperatureState']<20)
                $color="bg-primary";

            $returnTab.='<th class="'.$color.'">'.$value['temperatureState'].'</th>';
            if($hour[1]=="50"){
                $returnTab."</tr>";
            }
    }
    $returnTab.='</tr></tbody></table>';

    return $returnTab;
}
```

downloadFile.php

Listing 1.30 – ./web/model/downloadFile.php

```php
<?php

// grab the requested file's name
$fileName = filter_input(INPUT_GET, 'file', FILTER_SANITIZE_STRING);
$fileType = filter_input(INPUT_GET, 'fileType', FILTER_SANITIZE_STRING);
$name = filter_input(INPUT_GET, 'fileName', FILTER_SANITIZE_STRING);

$path="";

if($fileType=='img'){
    $path='graphs/';
    $name.='.png';
}

// make sure it's a file before doing anything!
if(file_exists($path.$fileName)) {

  header('Pragma: public');   // required
  header('Expires: 0');   // no cache
  header('Cache-Control: must-revalidate, post-check=0, pre-check=0');
  header('Cache-Control: private',false);
  header('Content-Type: image/png');
  header('Content-Disposition: attachment; filename="'.$name.'"');
  header('Content-Transfer-Encoding: binary');
  header('Content-Length: '.filesize($path.$fileName)); // provide file
      size
  header('Connection: close');
    readfile($path.$fileName);    // push it out
}

echo "<script>window.close();</script>";
exit();
```

findUserComputer.php

Listing 1.31 – ./web/model/findUserComputer.php

```php
<?php
/**
* @param int idUser
* @return string if true return the mac adress
* @return string null if false
*/
function getMacAdress($idUser){
    $connexion = getConnexion();

    $dayNumber = date('N');
    $dateOfTheDay = date('Y-m-d');
    $timeOfTheDay = date('H:i');

    $req = $connexion->prepare("
    SELECT computers.MAC
    FROM computers, users_has_Computer
    WHERE users_has_Computer.idUser = :idUser
    AND users_has_Computer.idComputer = computers.idComputer
    AND users_has_Computer.idLesson IN (SELECT lessons.idLesson
    FROM `lessons`, user_has_Lesson
    WHERE `dateDebut` <= :dateN
    AND `dateEnd` >= :dateN
    AND `weekDay` = :dayN
    AND `timeDebut` <= :timeN
    AND `timeEnd` >= :timeN
    AND user_has_Lesson.idUser = :idUser
    AND user_has_Lesson.idLesson = lessons.idLesson)
    ");

    $req->bindParam(":dateN", $dateOfTheDay, PDO::PARAM_STR);
    $req->bindParam(":timeN", $timeOfTheDay, PDO::PARAM_STR);
    $req->bindParam(":dayN", $dayNumber, PDO::PARAM_STR);
    $req->bindParam(":idUser", $idUser, PDO::PARAM_STR);

    $req->execute();
    $mac = $req->fetchAll(PDO::FETCH_ASSOC);
    if(count($mac)>0){
        return $mac[0]['MAC'];
    }
    return Null;
}
```

functions.php

Listing 1.32 – ./web/model/functions.php

```php
<?php
function randomPassword() {
    $alphabet = '
       abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890';
    $password = '';
    for ($i=0; $i < 5; $i++) {
        $rnd = rand(0, strlen($alphabet) - 1);
        $password .= $alphabet[$rnd];
    }
    return $password;
}

function readUser($filename){
    if (file_exists($filename)) {
        return file($filename);
    }
    return array('file doesn\'t exist');
}

/**
 * Return true if the date is between the dateDebut and the dateEnd else
    false
 * @param  DateTime dateDebut
 * @param  DateTime dateEnd
 * @param  DateTime date
 * @return Boolean
 */
function DateBetweenTowDate($dateDebut, $dateEnd, $date){
    return ($date >= $dateDebut) && ($date <= $dateEnd);
}
```

getTag.php

Listing 1.33 – ./web/model/getTag.php

```php
<?php
header('content-type:application/json');
require_once 'crud.php';
$tag = FILTER_INPUT(INPUT_GET, 'tagId', FILTER_SANITIZE_STRING);
require_once 'findUserComputer.php';
$information = Array();

if($tag != NULL){
    $idUser = checkTag($tag);
    array_push($information, ['Status' => '']);
    array_push($information, ['MacAdress' => '']);
    $information[1]['MacAdress'] = '00-00-00-00-00-00';
    if($idUser == NULL){
        $information[0]['Status']="error";
        array_push($information, [0 => 'Unknown User']);
    }else{
        $information[1]['MacAdress'] = getMacAdress($idUser[0]['idUser']);
        $userHaveClasses = didUserHaveClasses($idUser);

        if(!empty($information[1]['MacAdress'])){
            $lastLog = lastLogin($idUser[0]['idUser']);
            $lastLogout = lastLogout($idUser[0]['idUser']);
            if(!empty($lastLog)){
                $date1 = date_create($lastLog[0]['timeArrive']);
                $date2 = new DateTime('now');
                $diff = date_diff($date1, $date2);
                if($diff->h > 0 || $diff->i >= 5){
                    addDepart($lastLog[0]['idPresence']);
                    $information[0]['Status']="success";
                    array_push($information, [0 =>'Success']);
                }else{
                    $information[0]['Status']="error";
                    array_push($information, [0 => 'Less than 5 minutes'.
                        $diff->i]);
                }
            }else{
                if(!empty($lastLogout)){
                    $date1 = date_create($lastLogout[0]['timeDepart']);
                    $date2 = new DateTime('now');
                    $diff = date_diff($date1, $date2);
                    if($diff->h > 0 || $diff->i >= 5){
                        addArrive($idUser[0]['idUser']);
                        $information[0]['Status']="success";
                        array_push($information, [0 =>'Success']);
                    }else{
                        $information[0]['Status']="error";
                        array_push($information, [0 => 'Less than 5
                            minutes']);
                    }
                }else{
                    addArrive($idUser[0]['idUser']);
                $information[0]['Status']="success";
                array_push($information, [0 =>'Success']);
                }
            }
        }else{
            $information[0]['Status']="error";
            array_push($information, [0 => 'Student don\'t have classes'])
                ;
        }
    }
}else{
    $information[0]['Status']="error";
    array_push($information, [0=>'No tag id provided']);
}

echo json_encode($information);
```

graphOptionFormHandling.php

Listing 1.34 – ./web/model/graphOptionFormHandling.php

```
1 <?php
2
3 $startDate = filter_input(INPUT_POST, 'startDate', FILTER_SANITIZE_STRING)
    ;
4 $startTime = filter_input(INPUT_POST, 'startTime', FILTER_SANITIZE_STRING)
    ;
5 $endDate = filter_input(INPUT_POST, 'endDate', FILTER_SANITIZE_STRING);
6 $endTime = filter_input(INPUT_POST, 'endTime', FILTER_SANITIZE_STRING);
7 $classId = filter_input(INPUT_POST, 'classId', FILTER_SANITIZE_NUMBER_INT)
    ;
8 $submit = filter_input(INPUT_POST, 'submit', FILTER_SANITIZE_STRING);
```

interAbsCl.php

Listing 1.35 – ./web/model/interAbsCl.php

```php
<?php
require_once 'schedule.php';

$d = new DateTime(FILTER_INPUT(INPUT_GET, 'date', FILTER_SANITIZE_STRING))
    ;

$dateStr = $d->format('d.m.Y');
        $dayStr = $d->format('l');
$showPresence = "<h1>$dayStr $dateStr</h1>";

$class = getTeacherClass($_SESSION['idUser']);
if (!empty($class)) {
    $students = getClassStudents($class[0]['idClass']);
    $todayDate = new dateTime(date('Y-m-d'));


    foreach ($students as $s) {
    // var_dump($s);
    $schedule = getDayPersonSchedule($s['idUser'], $d->format('N'), $d->f
        ormat('Y-m-d'));
    $presAll = getDailyUserPresence($s['idUser'], $d->format('Y-m-d'));

        $strDateClass = $d->format('Y-m-d');
        $dailyDate = $d->format('l d.m.Y');

        $period = [
            'H1' => [
                'debut' => new DateTime($strDateClass . '07:30:00'),
                'end' => new DateTime($strDateClass . '08:00:00')
            ],
            'H2' => [
                'debut' => new DateTime($strDateClass . '08:05:00'),
                'end' => new DateTime($strDateClass . '08:50:00')
            ],
            'H3' => [
                'debut' => new DateTime($strDateClass . '08:55:00'),
                'end' => new DateTime($strDateClass . '09:40:00')
            ],
            'H4' => [
                'debut' => new DateTime($strDateClass . '10:05:00'),
                'end' => new DateTime($strDateClass . '10:50:00')
            ],
            'H5' => [
                'debut' => new DateTime($strDateClass . '10:55:00'),
                'end' => new DateTime($strDateClass . '11:40:00')
            ],
            'H6' => [
                'debut' => new DateTime($strDateClass . '11:45:00'),
                'end' => new DateTime($strDateClass . '12:30:00')
            ],
            'H7' => [
                'debut' => new DateTime($strDateClass . '12:40:00'),
                'end' => new DateTime($strDateClass . '13:25:00')
            ],
            'H8' => [
                'debut' => new DateTime($strDateClass . '13:30:00'),
                'end' => new DateTime($strDateClass . '14:15:00')
            ],
            'H9' => [
                'debut' => new DateTime($strDateClass . '14:35:00'),
                'end' => new DateTime($strDateClass . '15:20:00')
            ],
            'H10' => [
                'debut' => new DateTime($strDateClass . '15:25:00'),
                'end' => new DateTime($strDateClass . '16:10:00')
            ],
            'H11' => [
                'debut' => new DateTime($strDateClass . '16:15:00'),
```

```
                'end' => new DateTime($strDateClass . '16:45:00')
            ]
        ];

        $showPresence .= "<h5>";
        $showPresence .= $s['lastname'] . " " . $s['firstname'] . "</h5>";
        //If the person has classes in the morning
        if (count($schedule) > 0) {
            //Morning
            $tableMrn = createTable($period, $schedule[0]);
            $teacherMrn = getDailyUserPresence($schedule[0]['idUser'],
                $d->format('Y-m-d'));
            $mrnClass = $schedule[0]['nameLesson'];


            //If future date
            if ($d >= $todayDate) {
                $showPresence .= setHalfDay("Matin", $mrnClass, $tableMrn,
                    'N/A');

                if (count($schedule) > 1) {
                    $aftClass = $schedule[1]['nameLesson'];
                    $tableAftn = createTable($period, $schedule[1]);
                    $showPresence .= setHalfDay("AprÃ¨s-midi", $aftClass,
                        $tableAftn, 'N/A');
                }
            } else {
                //If teacher's abs
                if (empty($teacherMrn)) {
                    $showPresence .= setHalfDay("Matin", $mrnClass,
                        $tableMrn, 'CS');
                } else {

                    if (empty($presAll)) {
                        $showPresence .= setHalfDay("Matin", $mrnClass,
                            $tableMrn, 'â  ');
                        if (count($schedule) > 1) {
                            $showPresence .= setHalfDay("AprÃ¨s-midi",
                                $mrnClass, $tableMrn, 'â  ');
                        }
                    } else {
                        $morningClassesPeriods = getPeriods($period,
                            $schedule[0]);
                        $morningDiffStart = date_diff(new DateTime(
                            $presAll[0]['timeArrive']), $morningClas
                            sesPeriods[array_key_first($morningClas
                            sesPeriods)]['debut']);
                        $morningDiffEnd = date_diff(new DateTime($presAll
                            [0]['timeDepart']), $morningClassesPeriods[
                            array_key_last($morningClassesPeriods)]['end'])
                            ;

                        if (($morningDiffStart->format('%i') <= 5 && $mor
                            ningDiffStart->format('%h') == 0) && ($morningD
                            iffEnd->format('%i') <= 5 && $morningDiffEnd->f
                            ormat('%h') == 0) || ($morningDiffStart->invert
                             == 1 && $morningDiffEnd->invert == 1)) {
                            $showPresence .= setHalfDay("AprÃ¨s-midi",
                                $mrnClass, $tableMrn, 'â  ');
                        } else {
                            $time = $d->format('Y-m-d');

                            $showPresence .= <<<ABS
                            <table class="table table-dark">
                            <tr>
                            <td>Cours - Matin</td>
                            ABS;
                            foreach ($tableMrn as $key => $value) {
                                $showPresence .= $value;
                            }
                            $showPresence .= <<<MORNINGSCHE
                            </tr>
```

```
121                                 <tr>
122                                 <td>$mrnClass</td>
123                                 MORNINGSCHE;
124                                 foreach ($morningClassesPeriods as $key =>
                                        $pres) {
125                                     $morningDiffStart = date_diff($pres['debut
                                            '], new DateTime($presAll[0]['
                                            timeArrive']));
126                                     $morningDiffEnd = date_diff($pres['end'],
                                            new DateTime($presAll[0]['timeDepart'])
                                            );
127
128                                     if ((($morningDiffStart->format('%i') >
                                            20 && $morningDiffStart->format('%h')
                                            == 0) || $morningDiffStart->format('%h'
                                            ) > 0) && $morningDiffStart->invert ==
                                            0) || ($morningDiffEnd->format('%i') >
                                            20 && $morningDiffEnd->invert == 1))
129                                     $showPresence .= "<td>â  </td>";
130                                     elseif ($morningDiffStart->format('%i') >
                                            5 && $morningDiffStart->format('%h') ==
                                             0 && $morningDiffStart->invert == 0)
131                                     $showPresence .= "<td>â ((morningDiffStart->
```

$format('$showPresence .= "<td>â</td>";

$showPresence. = ” < /tr >< /table > ”;

//If the person has classes in the afternoon

if (count($schedule) > 1)

$aftClass$ =schedule[1]['nameLesson'];

$tableAftn = createTable($period, $schedule[1]);

$afternoonClassesPeriods = getPeriods($period, $schedule[1]);

$teacherAft = getDailyUserPresence(schedule[0]['idUser'], d- > format('Y - m - d'));

if (empty($teacherAft$))

$showPresence. = setHalfDay(”AprÃ¨s - midi”,aftClass, $tableAftn$,'CS');

else

if (count($presAll$) > 1)

$afternoonDiffStart = date_diff(newDateTime(presAll[1]['timeArrive']), afternoonClassesPerio

$afternoonDiffEnd = date_diff(newDateTime(presAll[1]['timeDepart']), afternoonClassesPerio

if (($afternoonDiffStart- > format('$showPresence .= setHalfDay("AprÃ¨s-midi", $aftClass$,table

'â');

else

$afternoonDiffStart = date_diff(newDateTime(presAll[0]['timeArrive']), afternoonClassesPerio

$afternoonDiffEnd = date_diff(newDateTime(presAll[0]['timeDepart']), afternoonClassesPerio

if (($afternoonDiffStart- > format('$showPresence .= setHalfDay("AprÃ¨s-midi", $aftClass$,table

'â');

if ($afternoonDiffStart- > format('$showPresence .= setHalfDay("AprÃ¨s-midi", $aftClass$,tableA

'â');

//If the person has classes in the afternoon and teacher abs

if (count($schedule) > 1)

$teacherAftn = getDailyUserPresence(schedule[1]['idUser'], d- > format('Y-m-d'));

$aftClass$ =schedule[1]['nameLesson'];

$tableAftn = createTable($period, $schedule[1]);

//If teacher's abs

if (empty($teacherAftn$))

$showPresence. = setHalfDay(”AprÃ¨s - midi”,aftClass, $tableAftn$,'CS');

else

$showPresence. =<<< MORNINGSCHE$

l'Ã©lÃ¨ve n'as pas cours

MORNINGSCHE;

```
    /**
    * @param array period is a table assoc with all the period the person has
    * @param array schedule is a table assoc with the schedule the person's in this day
    * @return array a table with the correct period in function of the schedule of the person
connected
    */
    function createTable(period,schedule)
    global strDateClass;
    table = array();
    foreach (periodaskey => value)
    intervalD = date_diff(value['debut'], new DateTime(strDateClass.schedule['timeDebut']));
    intervalE = date_diff(newDateTime(strDateClass . schedule['timeEnd']),value['end']);
    //Period between
    if ((intervalE- > format('table[key] = " < td >key</td>";
    //Period debut
    if (intervalD- > format('table[key] = " < td >key</td>";
    //Period end
    if (intervalE- > format('table[key] = " < td >key</td>";
    return table;
    /**
    * @param string define for the moment of the day
    * @param string class of the person
    * @return array period is a table assoc with all the period the person has
    * @return string show the status of the person during this period
    */
    function setHalfDay(time,class, periods,comment)
    showPresence =<<< ABS
    <table class="table table-dark">
    <tr>
    <td>Cours - time < /td >
    ABS;
    foreach (periodsaskey => value)
    showPresence. =value;
    showPresence. =<<< MORNINGSCHE
    </tr>
    <tr>
    <td>class < /td >
    MORNINGSCHE;
    foreach (periodsaskey => value)
    showPresence. = " < td >comment</td>";
    return showPresence;
    /**
    * @return array period is a table assoc with all the period the person has
    * @param array schedule is a table assoc with the schedule the person's in this day
    * @return array table assoc with all the period the person's during the day
    */
    function getPeriods(period,schedule)
    global strDateClass;
    table = array();
    foreach (periodaskey => value)
    intervalD = date_diff(value['debut'], new DateTime(strDateClass.schedule['timeDebut']));
    intervalE = date_diff(newDateTime(strDateClass . schedule['timeEnd']),value['end']);
```

```
//Period between
if ((intervalE-> format('table[key] =value;
//Period debut
if (intervalD-> format('table[key] =value;
//Period end
if (intervalE-> format('table[key] =value;
return table;
```

interAbsSt.php

Listing 1.36 – ./web/model/interAbsSt.php

```php
<?php
require_once 'schedule.php';

$d = new DateTime(FILTER_INPUT(INPUT_GET, 'date', FILTER_SANITIZE_STRING));
$schedule = getDayPersonSchedule($_SESSION['idUser'], $d->format('N'), $d->format('Y-m-d'));
$presAll = getDailyUserPresence($_SESSION['idUser'], $d->format('Y-m-d'));

$todayDate = new dateTime(date('Y-m-d'));

$strDateClass = $d->format('Y-m-d');
$dailyDate = $d->format('l d.m.Y');

$dateStr = $d->format('d.m.Y');
$dayStr = $d->format('l');

$period = [
    'H1' => [
        'debut' => new DateTime($strDateClass . '07:30:00'),
        'end' => new DateTime($strDateClass . '08:00:00')
    ],
    'H2' => [
        'debut' => new DateTime($strDateClass . '08:05:00'),
        'end' => new DateTime($strDateClass . '08:50:00')
    ],
    'H3' => [
        'debut' => new DateTime($strDateClass . '08:55:00'),
        'end' => new DateTime($strDateClass . '09:40:00')
    ],
    'H4' => [
        'debut' => new DateTime($strDateClass . '10:05:00'),
        'end' => new DateTime($strDateClass . '10:50:00')
    ],
    'H5' => [
        'debut' => new DateTime($strDateClass . '10:55:00'),
        'end' => new DateTime($strDateClass . '11:40:00')
    ],
    'H6' => [
        'debut' => new DateTime($strDateClass . '11:45:00'),
        'end' => new DateTime($strDateClass . '12:30:00')
    ],
    'H7' => [
        'debut' => new DateTime($strDateClass . '12:40:00'),
        'end' => new DateTime($strDateClass . '13:25:00')
    ],
    'H8' => [
        'debut' => new DateTime($strDateClass . '13:30:00'),
        'end' => new DateTime($strDateClass . '14:15:00')
    ],
    'H9' => [
        'debut' => new DateTime($strDateClass . '14:35:00'),
        'end' => new DateTime($strDateClass . '15:20:00')
    ],
    'H10' => [
        'debut' => new DateTime($strDateClass . '15:25:00'),
        'end' => new DateTime($strDateClass . '16:10:00')
    ],
    'H11' => [
        'debut' => new DateTime($strDateClass . '16:15:00'),
        'end' => new DateTime($strDateClass . '16:45:00')
    ]
];

$showPresence = "<h1>$dayStr $dateStr</h1>";

//If the person has classes in the morning
if (count($schedule) > 0) {
```

```
67      //Morning
68      $tableMrn = createTable($period, $schedule[0]);
69      $teacherMrn = getDailyUserPresence($schedule[0]['idUser'], $d->format(
            'Y-m-d'));
70      $mrnClass = $schedule[0]['nameLesson'];
71
72
73      //If future date
74      if ($d >= $todayDate) {
75          $showPresence .= setHalfDay("Matin", $mrnClass, $tableMrn, 'N/A');
76
77          if (count($schedule) > 1) {
78              $aftClass = $schedule[1]['nameLesson'];
79              $tableAftn = createTable($period, $schedule[1]);
80              $showPresence .= setHalfDay("AprÃ¨s-midi", $aftClass,
                    $tableAftn, 'N/A');
81          }
82      } else {
83          //If teacher's abs
84          if (empty($teacherMrn)) {
85              $showPresence .= setHalfDay("Matin", $mrnClass, $tableMrn, 'CS
                    ');
86          } else {
87
88              if (empty($presAll)) {
89                  $showPresence .= setHalfDay("Matin", $mrnClass, $tableMrn,
                        'â  ');
90                  if (count($schedule) > 1) {
91                      $showPresence .= setHalfDay("AprÃ¨s-midi", $mrnClass,
                            $tableMrn, 'â  ');
92                  }
93              } else {
94                  $morningClassesPeriods = getPeriods($period, $schedule[0])
                        ;
95                  $morningDiffStart = date_diff(new DateTime($presAll[0]['
                        timeArrive']), $morningClassesPeriods[array_key_first(
                        $morningClassesPeriods)]['debut']);
96                  $morningDiffEnd = date_diff(new DateTime($presAll[0]['
                        timeDepart']), $morningClassesPeriods[array_key_last($m
                        orningClassesPeriods)]['end']);
97
98                  if (($morningDiffStart->format('%i') <= 5 && $morningDif
                        fStart->format('%h') == 0) && ($morningDiffEnd->format(
                        '%i') <= 5 && $morningDiffEnd->format('%h') == 0) || (
                        $morningDiffStart->invert == 1 && $morningDif
                        fEnd->invert == 1)) {
99                      $showPresence .= setHalfDay("AprÃ¨s-midi", $mrnClass,
                            $tableMrn, 'â  ');
100                 } else {
101                     $time = $d->format('Y-m-d');
102
103                     $showPresence .= <<<ABS
104                     <table class="table table-dark">
105                     <tr>
106                     <td>Cours - Matin</td>
107                     ABS;
108                     foreach ($tableMrn as $key => $value) {
109                         $showPresence .= $value;
110                     }
111                     $showPresence .= <<<MORNINGSCHE
112                     </tr>
113                     <tr>
114                     <td>$mrnClass</td>
115                     MORNINGSCHE;
116                     foreach ($morningClassesPeriods as $key => $pres) {
117                         $morningDiffStart = date_diff($pres['debut'], new
                            DateTime($presAll[0]['timeArrive']));
118                         $morningDiffEnd = date_diff($pres['end'], new
                            DateTime($presAll[0]['timeDepart']));
119
120                         if ((($morningDiffStart->format('%i') > 20 && $m
                            orningDiffStart->format('%h') == 0) || $mor
```

```
                                ningDiffStart->format('%h') > 0) && $morningDif
                                fStart->invert == 0) || ($morningDiffEnd->for
                                mat('%i') > 20 && $morningDiffEnd->invert == 1)
                                )
121                              $showPresence .= "<td>â  </td>";
122                         elseif ($morningDiffStart->format('%i') > 5 && $m
                                orningDiffStart->format('%h') == 0 && $morningD
                                iffStart->invert == 0)
123                              $showPresence .= "<td>â ((morningDiffStart- > format('show
.= "<td>â</td>";
    //If the person has classes in the afternoon
    if (count(schedule) > 1)
    aftClass =schedule[1]['nameLesson'];
    tableAftn = createTable(period, schedule[1]);
    afternoonClassesPeriods = getPeriods(period, schedule[1]);
    teacherAft = getDailyUserPresence(schedule[0]['idUser'], d- > format('Y -m -d'));
    if(empty(teacherAft))
    showPresence. = setHalfDay("AprÃ¨s - midi",aftClass, tableAftn,'CS');
    else
    if (count(presAll) > 1)
    afternoonDiffStart = date_diff(newDateTime(presAll[1]['timeArrive']), afternoonClassesPerio
    afternoonDiffEnd = date_diff(newDateTime(presAll[1]['timeDepart']), afternoonClassesPerio
    if ((afternoonDiffStart- > format('showPresence .= setHalfDay("AprÃ¨s-midi", aftClass,table
'â');
    else
    afternoonDiffStart = date_diff(newDateTime(presAll[0]['timeArrive']), afternoonClassesPerio
    afternoonDiffEnd = date_diff(newDateTime(presAll[0]['timeDepart']), afternoonClassesPerio
    if ((afternoonDiffStart- > format('showPresence .= setHalfDay("AprÃ¨s-midi", aftClass,table
'â');
    if(afternoonDiffStart- > format('showPresence .= setHalfDay("AprÃ¨s-midi", aftClass,tableA
'â');
    //If the person has classes in the afternoon and teacher abs
    if (count(schedule) > 1)
    teacherAftn = getDailyUserPresence(schedule[1]['idUser'], d- > format('Y-m-d'));
    aftClass =schedule[1]['nameLesson'];
    tableAftn = createTable(period, schedule[1]);
    //If teacher's abs
    if (empty(teacherAftn))
    showPresence. = setHalfDay("AprÃ¨s - midi",aftClass, tableAftn,'CS');
    else
    showPresence =<<< MORNINGSCHE
    <h1>dailyDate < /h1 >
    <div class="alert alert-danger">Vous n'avez pas de cours Ã  cette date</div>
    MORNINGSCHE;
    /**
    * @param array period is a table assoc with all the period the person has
    * @param array schedule is a table assoc with the schedule the person's in this day
    * @return array a table with the correct period in function of the schedule of the person
connected
    */
    function createTable(period,schedule)
    global strDateClass;
    table = array();
    foreach (periodaskey => value)
```

```
intervalD = date_diff(value['debut'], new DateTime(strDateClass.schedule['timeDebut']));
intervalE = date_diff(newDateTime(strDateClass . schedule['timeEnd']),value['end']);
//Period between
if ((intervalE- > format('table[key] = " < td >key</td>";
//Period debut
if (intervalD- > format('table[key] = " < td >key</td>";
//Period end
if (intervalE- > format('table[key] = " < td >key</td>";
return table;
/**
* @param string define for the moment of the day
* @param string class of the person
* @return array period is a table assoc with all the period the person has
* @return string show the status of the person during this period
*/
function setHalfDay(time,class, periods,comment)
showPresence =<<< ABS
<table class="table table-dark">
<tr>
<td>Cours - time < /td >
ABS;
foreach (periodsaskey => value)
showPresence. =value;
showPresence. =<<< MORNINGSCHE
</tr>
<tr>
<td>class < /td >
MORNINGSCHE;
foreach (periodsaskey => value)
showPresence. = " < td >comment</td>";
return showPresence;
/**
* @return array period is a table assoc with all the period the person has
* @param array schedule is a table assoc with the schedule the person's in this day
* @return array table assoc with all the period the person's during the day
*/
function getPeriods(period,schedule)
global strDateClass;
table = array();
foreach (periodasskey => value)
intervalD = date_diff(value['debut'], new DateTime(strDateClass.schedule['timeDebut']));
intervalE = date_diff(newDateTime(strDateClass . schedule['timeEnd']),value['end']);
//Period between
if ((intervalE- > format('table[key] =value;
//Period debut
if (intervalD- > format('table[key] =value;
//Period end
if (intervalE- > format('table[key] =value;
return table;
```

login.php

Listing 1.37 – ./web/model/login.php

```php
<?php
$mail = FILTER_INPUT(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL);
$mdp = FILTER_INPUT(INPUT_POST, 'pwd', FILTER_SANITIZE_STRING);
$submit = FILTER_INPUT(INPUT_POST, 'submit', FILTER_SANITIZE_STRING);
$error = "";

if($submit){
    if(!FILTER_VAR($mail, FILTER_VALIDATE_EMAIL)){
        $error = '<div class="alert alert-danger mt-1" role="alert">Email non valide</div>';
    }else{
        if(login($mail, $mdp)){
            $_SESSION['log'] = $mail;
            $_SESSION['idUser'] = getIdUser($mail);
            header('Location: ?action=myAccount');
            exit;
        }else{
            $error = '<div class="alert alert-danger mt-1" role="alert">Email ou mot de passe non valide</div>';
        }
    }
}
```

logout.php

Listing 1.38 – ./web/model/logout.php

```
<?php
// DÃ©truit toutes les variables de session
$_SESSION = array();

// Si vous voulez dÃ©truire complÃ¨tement la session, effacez Ã©galement
// le cookie de session.
// Note : cela dÃ©truira la session et pas seulement les donnÃ©es de
    session !
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// Finalement, on dÃ©truit la session.
session_destroy();
header('Location: ?action=index');
exit;
```

myAccount.php

Listing 1.39 – `./web/model/myAccount.php`

removeUser.php

Listing 1.40 – ./web/model/removeUser.php

```php
<?php

$userId = filter_input(INPUT_GET, 'id', FILTER_SANITIZE_NUMBER_INT);

if(!empty($userId)){
    if(removeUser($userId)){
        header("Location: ?action=admin&done=remove&success=true");
        exit();
    }
    header("Location: ?action=admin&success=false");
    exit();
}else{
    header("Location: ?action=admin");
    exit();
}
```

schedule.php

Listing 1.41 – ./web/model/schedule.php

```php
<?php
$period = [
    'H1' => [
        'debut' => new DateTime('07:30:00'),
        'end' => new DateTime('08:00:00')
    ],
    'H2' => [
        'debut' => new DateTime('08:05:00'),
        'end' => new DateTime('08:50:00')
    ],
    'H3' => [
        'debut' => new DateTime('08:55:00'),
        'end' => new DateTime('09:40:00')
    ],
    'H4' => [
        'debut' => new DateTime('10:05:00'),
        'end' => new DateTime('10:50:00')
    ],
    'H5' => [
        'debut' => new DateTime('10:55:00'),
        'end' => new DateTime('11:40:00')
    ],
    'H6' => [
        'debut' => new DateTime('11:45:00'),
        'end' => new DateTime('12:30:00')
    ],
    'H7' => [
        'debut' => new DateTime('12:40:00'),
        'end' => new DateTime('13:25:00')
    ],
    'H8' => [
        'debut' => new DateTime('13:30:00'),
        'end' => new DateTime('14:15:00')
    ],
    'H9' => [
        'debut' => new DateTime('14:35:00'),
        'end' => new DateTime('15:20:00')
    ],
    'H10' => [
        'debut' => new DateTime('15:25:00'),
        'end' => new DateTime('16:10:00')
    ],
    'H11' => [
        'debut' => new DateTime('16:15:00'),
        'end' => new DateTime('16:45:00')
    ]
];

$nameLesson = FILTER_INPUT(INPUT_POST, 'namelesson',
    FILTER_SANITIZE_STRING);
$dateDebut = FILTER_INPUT(INPUT_POST, 'dateDebut', FILTER_SANITIZE_STRING)
    ;
$dateEnd = FILTER_INPUT(INPUT_POST, 'dateEnd', FILTER_SANITIZE_STRING);

$weekDay = FILTER_INPUT(INPUT_POST, 'weekDay', FILTER_VALIDATE_INT);

$timeDebut = FILTER_INPUT(INPUT_POST, 'timeDebut', FILTER_SANITIZE_STRING)
    ;
$timeEnd = FILTER_INPUT(INPUT_POST, 'timeEnd', FILTER_SANITIZE_STRING);

$idRoom = FILTER_INPUT(INPUT_POST, 'idRoom', FILTER_VALIDATE_INT);
$idUser = FILTER_INPUT(INPUT_POST, 'idUser', FILTER_VALIDATE_INT);

$submit = FILTER_INPUT(INPUT_POST, 'submit', FILTER_SANITIZE_STRING);

$RoomSelector = RoomSelector($idRoom);
$TeacherSelector = TeacherSelector($idUser);

```

```php
$timeDebutSelector = TimeSelector("Heure de debut", "timeDebut" , $timeDebut);
$timeEndSelector = TimeSelector("Heure de fin", "timeEnd" ,$timeEnd);

$error=[];
$validation = "";

$now = date("Y-m-d");
$dateMax = new DateTime($now);
$dateMax->add(new DateInterval('P2Y'));
$date1 = $dateMax;
$dateMax = $dateMax->format("Y-m-d");

if (isset($period[$timeDebut]) && isset($period[$timeEnd])){
    $timeDebut = $period[$timeDebut]["debut"];
    $timeEnd = $period[$timeEnd]["end"];

    if($submit){
    $noError = true;

        if ($nameLesson == ""){
            $noError = false;
            $error[] = "Veuillez saisir un nom valide!";
        }

        if (!DateBetweenTowDate($now, $dateMax, $dateDebut)){
            $noError = false;
            $error[] = "La date de debut n'est pas dans la limite dÃ©fini !";
        }
        if (!DateBetweenTowDate($now, $dateMax, $dateEnd)){
            $noError = false;
            $error[] = "La date de fin n'est pas dans la limite dÃ©fini !";
        }
        $date1 = new DateTime($dateDebut);
        $date2 = new DateTime($dateEnd);
        $interval = $date1->diff($date2);
        if ($interval->format('%a') < 30){
            $noError = false;
            $error[] = "L'interval entre la date de debut et de fin est inferieur Ã  30 jours!";
        }

        if (!(($weekDay >= 1) && ($weekDay <= 5))){
            $noError = false;
            $error[] = "Le nombre entrÃ© n'est pas entre 1 et 5 !";
        }

        if (!DateBetweenTowDate(new DateTime("07:00"), new DateTime("18:00"), $timeDebut)){
            $noError = false;
            $error[] = "L'heure de debut n'est pas dans la limite dÃ©fini !";
        }
        if (!DateBetweenTowDate(new DateTime("08:00"), new DateTime("19:00"), $timeEnd)){
            $noError = false;
            $error[] = "L'heure de fin n'est pas dans la limite dÃ©fini !";
        }
        $date1 = $timeDebut;
        $date2 = $timeEnd;
        $interval = $date1->diff($date2);
        if ($interval->format('%I') < 30 && $interval->format('%H') < 1){
            $noError = false;
            $error[] = "L'interval entre l'heure de debut et de fin est inferieur Ã  30 min!";
        }

```

```
127         if(count(getLessonsByIdRoomAndDateAndTime($idRoom, $dateDebut,
                $dateEnd, $weekDay, $timeDebut->format("H:i"), $timeEnd->format
                ("H:i"))) >= 1){
128             $noError = false;
129             $error[] = "Salle dÃ©jÃ  utilisÃ©e";
130         }
131         if (count(getLessonsByIdUserAndDateAndTime($idUser, $dateDebut,
                $dateEnd, $weekDay, $timeDebut->format("H:i"), $timeEnd->format
                ("H:i"))) >=1){
132             $noError = false;
133             $error[] = "Enseignant dÃ©jÃ  utilisÃ©";
134         }
135
136         if ($noError){
137             addLessons( $nameLesson, $dateDebut, $dateEnd, $weekDay,
                    $timeDebut->format("H:i"), $timeEnd->format("H:i"), $idRoom
                    , $idUser);
138             $validation = "Cours ajoutÃ©";
139         }
140     }
141 }
142
143 function RoomSelector($idRoomS){
144     $selector = "";
145     $rooms = getAllRoom();
146
147     $selector = '<div class="form-group"><label for="idRoom">Salle de clas
            se utilisÃ©e</label>';
148     $selector .= '<select name="idRoom" class="form-control" id="idRoom">'
            ;
149
150     foreach ($rooms as $key => $value) {
151         $idRoom = $value["idRoom"];
152         $nameRoom = $value["nameRoom"];
153         $selector .="<option value='$idRoom' ".($idRoom == $idRoomS ? '
                selected' : '') .">$nameRoom</option>";
154     }
155
156     $selector .= "</select></div>";
157     return $selector;
158 }
159
160 function TeacherSelector($idUserS){
161     $selector = "";
162     $teacher = getAllTeacher();
163
164     $selector = '<div class="form-group"><label for="idUser">Enseignant
            </label>';
165     $selector .= '<select name="idUser" class="form-control" id="idUser">'
            ;
166
167     foreach ($teacher as $key => $value) {
168         $idUser = $value["idUser"];
169         $name = $value["lastname"] . " " . $value["firstname"];
170         $selector .="<option value='$idUser' ".($idUser == $idUserS ? '
                selected' : '') .">$name</option>";
171     }
172
173     $selector .= "</select></div>";
174     return $selector;
175 }
176
177 function TimeSelector($text, $name ,$H){
178     $period = $GLOBALS["period"];
179
180     $selector = "";
181
182     $selector = '<div class="form-group"><label for="'.$name.'">'.$text.'
            </label>';
183     $selector .= '<select name="'.$name.'" class="form-control" id="'.
            $name.'">';
```

```
184
185     foreach ($period as $key => $value) {
186         $selector .="<option value='$key' ".($key == $H ? 'selected' : '')
                ."'>$key</option>";
187     }
188
189     $selector .= "</select></div>";
190     return $selector;
191 }
```

1.2.1 classes

card.php

Listing 1.42 – ./web/model/classes/card.php

```php
<?php

/**
 * @author Alexandre Benzonana
 * @version 1.0
 */
class Card extends GetterSetter{

    private $buildingName;
    private $classNumber;
    private $dateTime;
    private $temperature;
    private $cardSize;

    /**
     * @param string Building name
     * @param int Class number
     * @param string date and time yyyy-mm-dd hh:mm
     * @param double Temerature between -20 and 50
     * @param int Card size between 1 and 12
     */
    public function __construct($buildingName, $classNumber, $dateTime, $temperature, $cardSize){
        $this->set_BuildingName($buildingName);
        $this->set_ClassNumber($classNumber);
        $this->set_DateTime($dateTime);
        $this->set_Temperature($temperature);
        $this->set_CardSize($cardSize);
    }

    /**
     * @return string The card in bootstrap with all the card param
     */
    public function getCard(){
        $card = "<div class='card text-white bg-".$this->getCardType($this->temperature)." mt-2 mr-0 col-md-".$this->cardSize."'>";
        $card .= "<div class='card-header'>".$this->buildingName." Salle ".$this->classNumber."</div>";
        $card .= "<div class='card-body'><h5 class='card-title'>".$this->temperature;
        $card .= "</h5><p class='card-text'>".$this->dateTime."</p></div></div>";
        return $card;
    }


    /**
     * Get the keyWord color for the given temperature
     * @param double temperature
     * @return string keyWord for color in bootstrap
     */
    public function getCardType($temperature){
        $type = 'secondary';
        if($temperature >=20){
            $type = 'success';
        }elseif($temperature <16){
            $type='danger';
        }elseif($temperature<18){
            $type = 'warning';
        }elseif($temperature <20){
            $type='primary';
        }

        return $type;
    }

```

```
62      public function get_Temperature(){
63          return $this->temperature;
64      }
65      /**
66       * @param double Temperature between -20 and 50
67       */
68      public function set_Temperature($temperature){
69          if($temperature > -20 && $temperature < 50)
70              $this->temperature = $temperature;
71          else
72              throw new Exception('Invalid Temperature');
73      }
74
75      public function get_BuildingName(){
76          return $this->buildingName;
77      }
78      /**
79       * @param string the building name
80       */
81      public function set_BuildingName($buildingName){
82          $this->buildingName = $buildingName;
83      }
84
85      public function get_ClassNumber(){
86          return $this->temperature;
87      }
88      /**
89       * @param int class number
90       */
91      public function set_ClassNumber($classNumber){
92          $this->classNumber = $classNumber;
93      }
94
95      public function get_DateTime(){
96          return $this->dateTime;
97      }
98      /**
99       * @param string Date and time yyyy-mm-dd hh:mm
100      */
101     public function set_DateTime($dateTime){
102
103         //REGEX for the date with time
104         $re1='((?:(?:[1]{1}\\d{1}\\d{1}\\d{1})|(?:[2]{1}\\d{3}))[-:
                \\/.](?:[0]?[1-9]|[1][012])[-:\\/.](?:(?:[0-2]?\\d{1})
                |(?:[3][01]{1})))(?![\\d])';
105         $re2='(\\s+)';
106         $re3='((?:(?:[0-1][0-9])|(?:[2][0-3])|(?:[0-9])):(?:[0-5][0-9]))';
107
108         if (!preg_match_all ("/".$re1.$re2.$re3."/is", $dateTime))
109         {
110             throw new Exception("Invalid datetime format --> expected '
                    YYYY-MM-DD HH:MM'");
111         }else{
112             $this->dateTime = $dateTime;
113         }
114     }
115
116     public function get_CardSize(){
117         return $this->cardSize;
118     }
119     /**
120      * @param int the card size between 1 and 12
121      */
122     public function set_CardSize($cardSize){
123         if($cardSize > 0 && $cardSize < 13)
124             $this->cardSize = $cardSize;
125         else
126             throw new Exception('Invalid cardSize, expected number between
                    1 and 12');
127     }
128 }
```

GetterSetter.php

Listing 1.43 – ./web/model/classes/GetterSetter.php

```php
<?php
/**
 * @author Christoff Truter
 */
abstract class GetterSetter{
    // Check for property accessibility
  protected function Accessible($name)
  {
    // Check for available accessors
    if ((method_exists($this, "set_$name")) ||
      (method_exists($this, "get_$name")))
      return true;

    // Inform dev why access to a field was denied
    throw new Exception((property_exists($this, $name) == false)
          ? "Property $name does not exist"
          : "Property $name not accessible");
  }

  public function __get($name)
  {
    if ($this->Accessible($name))
    {
      // call get accessor
      if (method_exists($this, "get_$name"))
        return $this->{"get_$name"}();
      else
        throw new Exception("Writeonly Property $name");
    }
  }

  public function __set($name, $value)
  {
    if ($this->Accessible($name))
    {
      // call set accessor (if available)
      if (method_exists($this, "set_$name"))
        $this->{"set_$name"}($value);
      else
        throw new Exception("Readonly Property $name");
    }
  }
}
```

Image.php

Listing 1.44 – ./web/model/classes/Image.php

```php
<?php

/**
 * @author Alexandre Benzonana
 * @version 1.2
 */
class Image extends GetterSetter{

    private const GREEN_MIN = 20;
    private const YELLOW_MAX = 20;
    private const ORANGW_MAX = 18;
    private const RED_MAX = 16;

    private const LEFT_MARGIN = 100;
    private const RIGHT_MARGIN = 50;
    private const TOP_MARGIN = 50; //Min 30
    private const BOTTOM_MARGIN = 70;

    private const TEMP_SPACE = 50;
    private const TIME_SPACE = 120;

    private const FONT_SIZE = 15;

    private $height;
    private $width;
    private $image;
    private $savePath;
    private $backColor;

    /**
     *
     * @param int with of the result image
     * @param int height of the result image
     */
    public function __construct(int $height, int $width, string $savePath,
        string $backColor="60, 60, 60"){
        $this->set_Width($width);
        $this->set_Height($height);
        $this->set_SavePath($savePath);
        $this->set_BackColor($backColor);
    }

    /**
     *
     */
    public function drawGraph(array $temperatures, $classInfo){

        $image = imagecreate($this->width, $this->height);
        $color = $this->backColor;
        imagecolorallocate($image, $color[0], $color[1], $color[2]);

        $white = imagecolorallocate($image, 250, 250, 250);
        $green = imagecolorallocate($image, 30, 100, 20);
        $yellow = imagecolorallocate($image, 198, 88, 0);
        $orange = imagecolorallocate($image, 255, 170, 0);
        $red = imagecolorallocate($image, 250, 50, 50);

        $graphHeight = $this->height - ($this::TOP_MARGIN +
            $this::BOTTOM_MARGIN);
        $graphWidth = $this->width - ($this::LEFT_MARGIN +
            $this::RIGHT_MARGIN);
        $graph00 = ['x'=>$this::LEFT_MARGIN, 'y'=>$this->height -
            $this::BOTTOM_MARGIN];

        $nbGraphTimeValues = round(($graphWidth)/$this::TIME_SPACE, 0);
        $nbTempValues = round(($graphHeight)/$this::TEMP_SPACE, 0);

        //On windows
```

```
65          // $font = "Arial.otf";
66          //On linux
67          $font = "/mnt/d/Ecole/ProjetsWeb/ARL/ARL/web/Arial.otf"; //Chemin
                vers le fichier Arial.otf (PrÃ©sent dans le dossier web du
                projet)
68          $fontSize = $this::FONT_SIZE;
69
70          $uniqueId = md5(rand(-99999,999999).json_encode($temperatures).r
                and(-999999,999999));
71
72          $minDateValue = min($temperatures);
73          $maxDateValue = max($temperatures);
74
75          $minTempValue = 50;
76          $maxTempValue = -50;
77
78          foreach($temperatures as $value){
79              if($value['temperatureState']<$minTempValue)
80                  $minTempValue=$value['temperatureState'];
81
82              if($value['temperatureState']>$maxTempValue)
83                  $maxTempValue=$value['temperatureState'];
84          }
85
86          $valueNumber = count($temperatures);
87
88          $tempGap = abs($maxTempValue-$minTempValue)/($nbTempValues);
89
90          $interval = date_diff(new DateTime($minDateValue['dateState']),
                new DateTime($maxDateValue['dateState']));
91
92          $timeJump = $valueNumber/$nbGraphTimeValues;
93
94          //Write graph time interval and class on the top left
95          imagettftext($image, $fontSize, 0, 10, $fontSize+5, $white, $font,
                $minDateValue['dateState'].' - '.$maxDateValue['dateState'] .
                ' -> Classe ' . $classInfo['className'].' '.$classInfo['
                buildingName']);
96          //Legend
97          imagettftext($image, $fontSize, 0, $this::LEFT_MARGIN/10,
                $this->height-$this::BOTTOM_MARGIN/1.7, $white, $font, "Â°C/
                Time");
98
99          //draw axis
100         for($i = 0; $i < 5; $i++){
101             imageline($image, $graph00['x']-25, $graph00['y']+$i+1,
                    $graph00['x']+$graphWidth+10, $graph00['y']+$i+1, $white);
102             imageline($image, $graph00['x']-$i-1, $graph00['y']+25,
                    $graph00['x']-$i-1, $graph00['y']-$graphHeight-10, $white);
103         }
104
105         //Write Temperatures and Temp landmark
106         $iteration=0;
107         $posY18 = $graph00['y']-30-(18-$minTempValue)*($graphHeight-30)/(
                $maxTempValue-$minTempValue);
108         $posY0 = $graph00['y']-30-(0-$minTempValue)*($graphHeight-30)/(
                $maxTempValue-$minTempValue);
109
110         for($i = $minTempValue; (double)$i <=(double)$maxTempValue
                +0.00001; $i+=$tempGap){
111
112             $posY=$graph00['y']-30-($i-$minTempValue)*($graphHeight-30)/(
                    $maxTempValue-$minTempValue);
113
114             if(abs($posY - $posY18)>$this::TEMP_SPACE/3 && abs($posY -
                    $posY0)>$tempGap/3){
115
116                 imagettftext($image, $fontSize, 0, $this::LEFT_MARGIN/2.5,
                        $posY+$fontSize/2, $white, $font, round($i,1));
117
118                 for($j = $graph00['x']; $j < $this::LEFT_MARGIN +
                        $graphWidth+10; $j += 20){
```

```
                    imageline($image, $j, $posY, $j + 5, $posY, $white);
                }
            }
            $iteration++;
        }

        //Write Date/Time and Time landmark
        $iteration=0;
        for($i = 0; $i<$valueNumber; $i+=ceil($timeJump)){

            $posX =$graph00['x']+60+$iteration*$this::TIME_SPACE;

            for($j = $graph00['y']; $j > $this::TOP_MARGIN-10; $j -= 20){
                imageline($image, $posX, $j, $posX, $j+5, $white);
            }

            $dateTime = explode(" ", $temperatures[ceil($i)]['dateState']);
            $date = $dateTime[0];
            $time = explode(":",$dateTime[1])[0].":".explode(":",$dateTime[1])[1];

            imagettftext($image, $fontSize, 0, $posX-$this::TIME_SPACE/($this::TIME_SPACE/25), $graph00['y']+30, $white, $font, $time);
            imagettftext($image, $fontSize, 0, $posX-$this::TIME_SPACE/($this::TIME_SPACE/50), $graph00['y']+50, $white, $font, $date);
            $iteration++;
        }

        //Draw warning landMark for 18Â°C and 16Â°C
        if($minTempValue <= 18){
            for($i = $graph00['x']/2; $i<$graphWidth+$graph00['x']; $i+=30){
                    imageline($image, $i, $posY18, $i+15, $posY18, $red);
            }
            imagettftext($image, 15, 0, $this::LEFT_MARGIN/8, $posY18+$fontSize/2, $orange, $font,"18");

            if($minTempValue <= 0){
                for($i = 40; $i<0.99*$width; $i+=20){
                    imageline($image, $i, $posY0, $i+5, $posY0, $lightRed);
                }
                imagettftext($image, 15, 0, 10, $posY0+$fontSize/2, $red, $font,"0");
            }
        }

        $posX = $graph00['x']+60;
        for($i=0; $i < $valueNumber - 1; $i++){

            $posY = $graph00['y']-30-($temperatures[$i]['temperatureState']-$minTempValue)*($graphHeight-30)/($maxTempValue-$minTempValue);
            $posY2 = $graph00['y']-30-($temperatures[$i+1]['temperatureState']-$minTempValue)*($graphHeight-30)/($maxTempValue-$minTempValue);

            if($temperatures[$i]['temperatureState']<16 || $temperatures[$i+1]['temperatureState']<16 ){
                $color=$red;
            }elseif($temperatures[$i]['temperatureState']<18 || $temperatures[$i+1]['temperatureState']<18){
                $color=$yellow;
            }elseif($temperatures[$i]['temperatureState']<20 || $temperatures[$i+1]['temperatureState']<20){
                $color=$orange;
            }else{
                $color=$green;
            }
```

```
174             imageline($image, $posX, $posY-1, $posX+($graphWidth-60)/
                    $valueNumber, $posY2-1, $color);
175             imageline($image, $posX, $posY, $posX+($graphWidth-60)/
                    $valueNumber, $posY2, $color);
176             imageline($image, $posX, $posY+1, $posX+($graphWidth-60)/
                    $valueNumber, $posY2+1, $color);
177
178             // if($i%ceil($timeJump)==0){
179             //     $dateTime = explode(" ", $temperatures[$i]['dateState'
                    ]);
180             //     $date = $dateTime[0];
181             //     $time = explode(":",$dateTime[1])[0].":".explode(":",
                    $dateTime[1])[1];
182
183             //     imagettftext($image, $fontSize, 0,
                    $posX-$this::TIME_SPACE/($this::TIME_SPACE/25), $graph00['y
                    ']+30, $white, $font, $time);
184             //     imagettftext($image, $fontSize, 0,
                    $posX-$this::TIME_SPACE/($this::TIME_SPACE/50), $graph00['y
                    ']+50, $white, $font, $date);
185             // }
186             $posX+=($graphWidth-60)/$valueNumber;
187         }
188
189         //Store the final image
190         if(imagepng($image,"graphs/".$uniqueId.".png")){
191             storeGraph($minDateValue['dateState'], $maxDateValue['
                    dateState'], "graphs/".$uniqueId.".png", $temperatures[0]['
                    idClass']);
192             return "graphs/".$uniqueId.".png";
193         }
194         return false;
195     }
196
197     public function getImage(){
198         return $this->$image;
199     }
200
201     public function get_Height(){
202         return $this->height;
203     }
204     /**
205      * @param int height min 400
206      */
207     public function set_Height(int $height){
208         if($height >= 400)
209             $this->height = $height;
210         else
211             $this->height = 400;
212     }
213
214     public function get_Width(){
215         return $this->width;
216     }
217
218     /**
219      * @param int widht min 800
220      */
221     public function set_Width(int $width){
222         if($width >= 800)
223             $this->width = $width;
224         else
225             $this->width = 800;
226     }
227
228     public function get_SavePath(){
229         return $this->savePath;
230     }
231
232     /**
233      * @param string path where the image have to be saved
234      */
```

```
235     public function set_SavePath(string $savePath){
236         $this->savePath = $savePath;
237     }
238 
239     public function get_BackColor(){
240         return $this->$backColor;
241     }
242 
243     /**
244      * @param string background color for the image, format "r, g, b"
245      */
246     public function set_BackColor(string $color){
247         $color = explode(',', $color);
248 
249         if(count($color)!=3){
250             throw new Exception('Invalid color, expected format "r, g, b"
                    ');
251             return;
252         }elseif($color[0] < 0 || $color[0] > 255 || $color[1] < 0 || $col
                or[1] > 255 || $color[2] < 0 || $color[2] > 255){
253             throw new Exception('Invalid color, value have to be between 0
                     and 255');
254             return;
255         }
256         $color = array_map('trim', $color);
257         $this->backColor = $color;
258     }
259 
260 }
```