

Information Systems and Databases

Report 3rd Project Assignment

Health care center database

29/11/2017

1st Semester – 2017/2018

Professors: Mário Gaspar da Silva
Francisco Regateiro
Miguel Amaral

Group number 49:

81731 – Catarina Aleixo
81138 – João Ramiro
81567 – José Miragaia

I. Introduction

This project consists in a Web-based application to perform some tasks that deal with the database that was populated in the 2nd assignment.

In order to unify the four tasks that will be explained in the following sections, it was created a home page (homepage.php) that has all the possibilities that users have to interact with the health care centre, that is represented in Figure 1.

It was created a php file that contains only the code for connecting to the database, to avoid repeating it in all files, that is called connectDB.php.

- [connectDB.php:](#)

```
<html>
  <body>
    <?php
      $host = "db.tecnico.ulisboa.pt";
      $user = "ist181731";
      $pass = "ahcu2726";
      $dsn = "mysql:host=$host;dbname=$user";
      try
      {
          $connection = new PDO($dsn, $user, $pass);
      }
      catch(PDOException $exception)
      {
          echo("<p> Error: ");
          echo($exception->getMessage());
          echo("</p>");
          exit();
      }
    ?>
  </body>
</html>
```

- [homePage.php:](#)

```
<html>
  <body>
    <h1><strong><font color= '#66CC00'>Welcome to the Healthcare Centre</font></strong></h1>
    <hr/>
    <form action="getPatient.php" method="post">
      <p>Search for/Insert a patient: <input type="submit" value="Search"/></p>
    </form>
    <form action="newStudy.php" method="post">
      <p>Create a study: <input type="submit" value="New Study"/></p>
    </form>
    <form action="getRegion.php" method="post">
      <p>Add a region to an element: <input type="submit" value="New Region"/></p>
    </form>
  </body>
</html>
```



Figure 1 – Home Page

II. Check patient existence

The first task consists of searching for a patient, given the full name or a portion of it. If the patient doesn't exist, then the user can register a new patient, giving the patient name, birthday and address.

The page to search for a patient will be displayed when the user clicks on the search button in the home page. The next page it's the `getPatient.php` that has a form that asks for the name the user wants to look for, as it can be seen in Figure 2.

- `getPatient.php`:

```
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Search for a Patient</font></strong></h3>
    <hr/>
    <form action="checkExistance.php" method="post">
      <p>Patient Name:
        <input type="text" name="name" require/>
        <input type="submit" value="Search"/>
      </p>
    </form>
    <?php
      echo("<p> Turn to the <a href='\"homePage.php\"'>home page</a>")
    ?>
  </body>
</html>
```



Figure 2 – Search for a patient

As soon as the user presses the Search button, then the `checkExistance.php` is executed. In this file, with the portion of the name given, it will be performed a query to search for all the patients that have names that match with the given substring and it is presented a table with all the results and a possibility of registering a new patient. Either if it returns a patient or not, it was chosen to give that possibility of registering a new patient as it can be seen in Figure 3.

- `checkExistance.php`:

```
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Check Patient Existence</font></strong></h3>
    <hr/>
    <?php
      if(($_REQUEST['name']))
      {
        require 'connectDB.php';    /* Connect to database */

        $name = $_REQUEST['name']; /* Get the portion of name that was searched */
        $findName = "%" . $name . "%";
        /* Prepare query to search for patients that have a name with that substring */
        $stmt = $connection->prepare("SELECT * FROM Patient WHERE name LIKE :findName");
```

```

$stmt->bindParam(':findName', $findName);
$stmt->execute();

if($stmt == FALSE)
{
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}

$nrows= $stmt->rowCount();
/* If records were found */
if($nrows > 0)
{
    echo("<table border=\"0\" cellspacing=\"5\" bordercolor='#66CC00'>");
    echo("<tr><td bgcolor='#66CC00'><font color='FFFFFF'>Patient Number</font></td><td bgcolor='#66CC00'><font color='FFFFFF'>Patient Name</font></td><td bgcolor='#66CC00'><font color='FFFFFF'>Birthday</font></td><td bgcolor='#66CC00'><font color='FFFFFF'>Address</font></td></tr>");

    foreach($stmt as $row)
    {
        echo("<tr><td>");
        echo("{ $row['patient_number']}");
        echo("</td><td>");
        echo("<a href='\"getDevices.php?patient_number=$patient_number(\"{$row['name']})\"</a>");
        echo("</td><td>");
        echo("{ $row['patient_number']}");
        echo("</td><td>");
        echo("{ $row['patient_number']}");
        echo("</td></tr>");
    }
    echo("</table>");
}
else
{
    echo("<p>Patient not found!</p>");
}
/* Possibility of registering a new patient */
?>
<form action="getNewPatient.php" method="post">
<p><input type="submit" value="Register a New Patient"/></p>
</form>
<?php
    echo("<p>Turn to the <a href='\"getPatient.php\">previous page</a></p>");
    $connection = null;
}
else
{
    echo("<p>No patient was searched</p>");
    echo("<p><a href='\"getPatient.php\">Search for a patient</a></p>");
}

?>
</body>
</html>

```

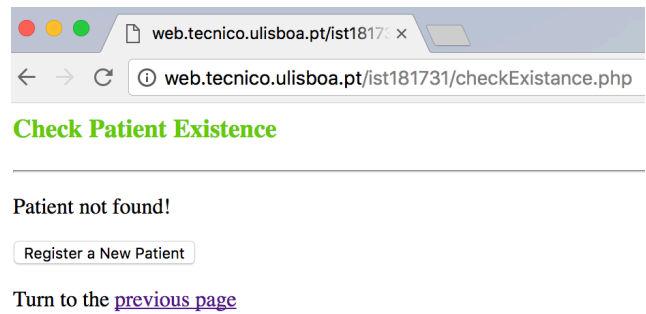
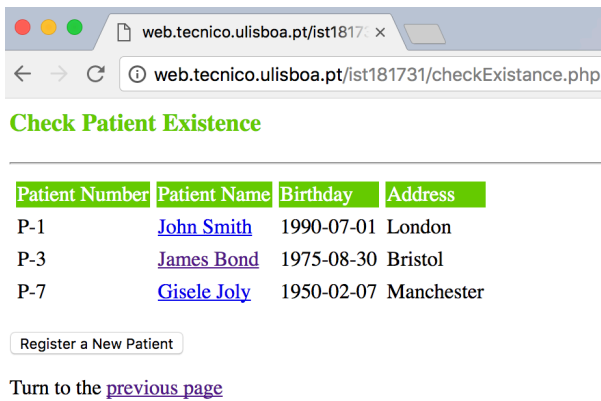


Figure 3 – Display records that contain the substring searched

In order to register a new patient, the user has to press the button Register New Patient. Otherwise, to make another search, the user can press the link to return to the previous page.

In the first case, it will be displayed a web page that contains a form (getNewPatient.php) with three fields: the name of the new patient, birthday and address as represented in Figure 4.

- [getNewPatient.php:](#)

```
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Register New Patient</font></strong></h3>
    <hr/>
    <form action="insertPatient.php" method="post">
      <p>Name: <input type="text" name="newName" required/></p>
      <p>Birthday: <input type="text" name="newBirthday" required/></p>
      <p>Address: <input type="text" name="newAddress" required/></p>
      <p><input type="submit" value="Register"/></p>
    </form>
  </body>
</html>
```



Figure 4 – Register a new patient

When the register button it's pressed, then it will be displayed a web page that confirms the registration (Figure 5). The changes in the database are performed in a file called insertPatient.php.

- [insertPatient.php:](#)

```
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Confirm Registration</font></strong></h3>
    <hr/>
    <?php
```

```

if(isset($_REQUEST['newName']) && isset($_REQUEST['newBirthday']) && isset($_REQUEST['newAddress']))
{
    if($_REQUEST['newBirthday'] > date("Y-m-d"))
    {
        echo("<p>Please enter a valid birthday date</p>");
        echo("<p>Turn to the <a href=\"getNewPatient.php\">previous page</a></p>");
        exit();
    }

    require 'connectDB.php';

    $sql = "SELECT patient_number FROM Patient";
    $result = $connection->query($sql);

    if($result == FALSE)
    {
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    /* Assign the next number, of the last one that exists, to the new patient */
    $patient_number = "P-" . (($result->rowCount()+1);

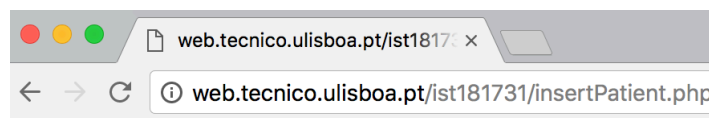
    $name = $_REQUEST['newName'];
    $birthday = $_REQUEST['newBirthday'];
    $address = $_REQUEST['newAddress'];

    /* Prepare against SQL injection */
    $stmt = $connection->prepare("INSERT INTO Patient VALUES (:patient_number, :name, :birthday, :address)");
    $stmt->bindParam(':patient_number', $patient_number);
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':birthday', $birthday);
    $stmt->bindParam(':address', $address);
    $stmt->execute();

    if($stmt == FALSE)
    {
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    echo("<p>Patient registered!</p>");
    echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");
}
else
{
    echo("<p>No patient was searched</p>");
    echo("<p><a href=\"getPatient.php\">Search for a patient</a></p>");
}
$connection = null;
?>
</body>
</html>

```



Confirm Registration

Patient registered!

Turn to the [Home page](#)

Figure 5 – Confirm registration

III. Replace Devices

When the user searches for a certain patient, the result table has a link in the field name. That link (represented in Figure 3) goes to another web page (getDevices.php) that shows not only the devices currently attached to that patient, but also the ones that the patient wore, since the patient was registered in the health care centre, as it can be seen in Figure 6.

- [getDevices.php:](#)

```
<?php session_start(); ?>
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Devices</font></strong></h3>
    <hr/>
    <?php
      if(isset($_GET['patient_number']))
      {
        require 'connectDB.php';
        $patient_number = $_GET['patient_number'];
        $findName = $_GET['findName'];
        $_SESSION['patient_number'] = $patient_number;

        $stmt = $connection->prepare("SELECT * FROM Wears WHERE patient_number=:patient_number ORDER BY end_date DESC");
        $stmt->bindParam(':patient_number', $patient_number);
        $stmt->execute();

        if($stmt == FALSE)
        {
          $info = $connection->errorInfo();
          echo("<p>Error: {$info[2]}</p>");
          exit();
        }

        $nrows= $stmt->rowCount();
        if($nrows > 0)
        {
          echo("<table border=\"0\" cellspacing=\"5\" bordercolor='#66CC00'>");
          echo("<tr><td bgcolor='#66CC00'><font color= 'FFFFFF'>Serial Number</font></td><td bgcolor='#66CC00'><font color= 'FFFFFF'>Manufacturer</font></td><td bgcolor='#66CC00'><font color= 'FFFFFF'>Start Date</font></td><td bgcolor='#66CC00'><font color= 'FFFFFF'>End Date</font></td></tr>");

          foreach($stmt as $row)
          {
            if($row['end_date'] >= date("Y-m-d H:i:s"))
            {
              echo("<tr><td><strong>");
              echo($row['serialnum']);
              echo("</strong></td><td><strong>");
              echo($row['manufacturer']);
              echo("</strong></td><td><strong>");
              echo($row['start_date']);
              echo("</strong></td><td><strong>");
              echo($row['end_date']);
              echo("</strong></td><td>");
            }
          }
        }
      }
    </?php>
  </body>
</html>
```

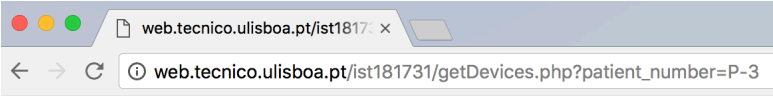
```

        echo("<form action=\"replaceDevices.php\" method=\"get\">");
        echo("<input type=\"hidden\" name=\"serialnum\" value=\"{$row['serialnum']}\"/>");
        echo("<input type=\"hidden\" name=\"manufacturer\" value=\"{$row['manufacturer']}\"/>");
        echo("<input type=\"hidden\" name=\"start_date\" value=\"{$row['start_date']}\"/>");
        echo("<input type=\"hidden\" name=\"end_date\" value=\"{$row['end_date']}\"/>");
        echo("<input type=\"submit\" value=\"Replace\"/>");
        echo("</form>");
        echo("</td></tr>");
    }
    else
    {
        echo("<tr><td>");
        echo($row['serialnum']);
        echo("</td><td>");
        echo($row['manufacturer']);
        echo("</td><td>");
        echo($row['start_date']);
        echo("</td><td>");
        echo($row['end_date']);
        echo("</td></tr>");
    }
}

echo("</table>");
echo("<p>Turn to the <a href=\"checkExistence.php?findName=$findName\">previous page</a></p>");
}
else
{
    echo("<p>Patient never wore any device!</p>");
    echo("<p>Turn to the <a href=\"checkExistence.php?findName=$findName\">previous page</a></p>");
}
}
else
{
    echo("<p>No patient was searched</p>");
    echo("<p><a href=\"getPatient.php\">Search for a patient</a></p>");
}

$connection = null;
?>
</body>
</html>

```



Serial Number	Manufacturer	Start Date	End Date	
A1	AirSense	2016-01-02 10:08:00	2030-12-31 00:00:00	Replace
H2	Honeywell	2017-07-31 00:00:00	2030-12-31 00:00:00	Replace

Figure 6 – Devices currently attached to a patient

Each device the patient is currently wearing has the possibility of being changed by another one of the same manufacturer that is available (replaceDevices.php). The replacement is done by selecting one of the possible devices that are in a drop-down list and by pushing a button called Replace, as represented in Figure 7.

- [replaceDevices.php:](#)

```

<?php session_start(); ?>
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Replace Device</font></strong></h3>
    <hr/>
    <form action="insertReplace.php" method="post">
      <?php
        if(isset($_GET['serialnum']) && isset($_GET['manufacturer']) && isset($_SESSION['patient_number']) &&
        isset($_GET['start_date']) && isset($_GET['end_date']))
        {
          require 'connectDB.php';
          $serialnum = $_GET['serialnum'];
          $manufacturer = $_GET['manufacturer'];
          $patient_number = $_SESSION['patient_number'];
          $end_date = $_GET['end_date'];
          $_SESSION['serialnum'] = $serialnum;
          $_SESSION['manufacturer'] = $manufacturer;
          $_SESSION['end_date'] = $end_date;
          $_SESSION['start_date'] = $_GET['start_date'];

          $stmt = $connection->prepare("SELECT serialnum FROM Device WHERE serialnum <> :serialnum AND
          manufacturer = :manufacturer AND serialnum NOT IN(SELECT serialnum FROM Wears WHERE serialnum <> :serialnum AND
          manufacturer = :manufacturer AND TIMESTAMPDIFF(SECOND, end_date, CURRENT_TIMESTAMP()) <= 0)");
          $stmt->bindParam(':serialnum', $serialnum);
          $stmt->bindParam(':manufacturer', $manufacturer);
          $stmt->execute();

          if($stmt == FALSE)
          {
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
          }

          $nrows= $stmt->rowCount();

          if($nrows > 0)
          {
            ?>
            <p>Devices available:
            <select name="serialnum">
              <?php
                foreach($stmt as $row)
                {
                  $serialnum=$row['serialnum'];
                  echo("<option value=\"{$serialnum}\">{$serialnum}</option>");
                }
              ?>
            </select>
            </p>
            <input type="submit" value="Replace"/>
            <?php

```

```

    }
    else
    {
        echo("<p>No devices of that manufacturer are available at the current time!</p>");
        echo("<p>Turn to the <a href='\"getDevices.php?patient_number=$patient_number\"'>previous
page</a></p>");
    }
}
else
{
    echo("<p>No patient was searched</p>");
    echo("<p><a href='\"getPatient.php\"'>Search for a patient</a></p>");
}

$connection = null;

?>

</form>
</body>
</html>

```

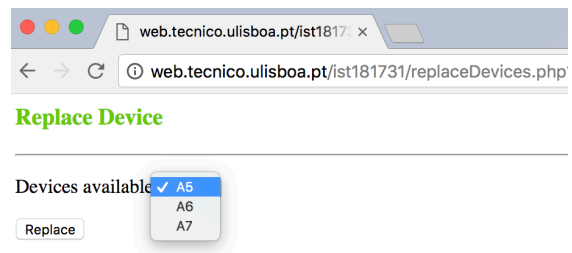


Figure 7 – Devices available to replace

In the insertReplace.php file the first thing that has to be done is checking if both of the new periods already exist in table Period. In order to clarify which periods have to be inserted, let's consider two devices A and B, in which A is being replaced by device B. So, the first period that has to be checked is the period that is going to be used to update the record about device A in table Wears. If the replace is being done today, the period is [start_date of device A, today]. The other period is the one that it's needed to insert the new record in table Wears about device B that is [today, end_date of device A]. If the periods don't exist, then they will be inserted.

After that, it will be inserted a new record and updated another one in table Wears, which corresponds to updating record A with the new period [start_date of device A, today] and inserting a new record for device B with period [today, end_date of device A]. If the update was a success, then a message with a confirmation of the replacement it will be displayed (Figure 8).

- [insertReplace.php:](#)

```

<?php session_start(); ?>
<html>
    <body>
        <h3><strong><font color= '#66CC00'>Replacement Confirmation</font></strong></h3>
        <hr/>
        <?php
            if(isset($_SESSION['patient_number']) && isset($_SESSION['serialnum']) && isset($_SESSION['manufacturer']) &&
            isset($_SESSION['start_date']) && isset($_SESSION['end_date']))
        {
            require 'connectDB.php';

```

```

$connection->beginTransaction();

$patient_number = $_SESSION['patient_number'];
$oldSerialNum = $_SESSION['serialnum'];
$oldManufacturer = $_SESSION['manufacturer'];
$oldStartDate = $_SESSION['start_date'];
$oldEndDate = $_SESSION['end_date'];
$currentTimeDate = date("Y-m-d H:i:s");

$sql = "SELECT * FROM Period";
$result = $connection->query($sql);

if($result == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}
$newPeriodExists = 0; /* To check if the period of the new current device exists */
$updatePeriodExists = 0; /* To check if the period of the replaced device exists */

foreach($result as $row)
{
    if($currentTimeDate == $row['start_date'] && $oldEndDate == $row['end_date'])
    {
        $newPeriodExists = 1;
    }
}

if($newPeriodExists == 0)
{
    $stmt = $connection->prepare("INSERT INTO Period VALUES (:start_date, :end_date)");
    $stmt->bindParam(':start_date', $currentTimeDate);
    $stmt->bindParam(':end_date', $oldEndDate);
    $stmt->execute();

    if($stmt == FALSE)
    {
        $connection->rollback();
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }
}

foreach($result as $row)
{
    if($oldStartDate == $row['start_date'] && $currentTimeDate = $row['end_date'])
    {
        $updatePeriodExists = 1;
    }
}

if($updatePeriodExists == 0)
{
    $stmt = $connection->prepare("INSERT INTO Period VALUES (:start_date, :end_date)");
    $stmt->bindParam(':start_date', $oldStartDate);
    $stmt->bindParam(':end_date', $currentTimeDate);
    $stmt->execute();
    if($stmt == FALSE)
    {
        $connection->rollback();
    }
}

```

```

        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }
}

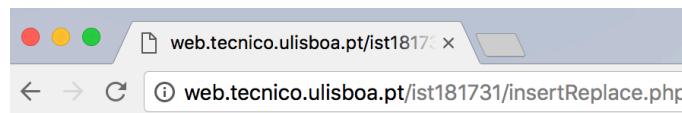
/* Updates the end date of the replaced device to the current date */
$stmt = $connection->prepare("UPDATE Wears SET start_date = :oldStartDate, end_date =
:currentTimeDate WHERE patient_number = :patient_number AND start_date = :oldStartDate AND end_date =
:oldEndDate");
$stmt->bindParam(':currentTimeDate', $currentTimeDate);
$stmt->bindParam(':patient_number', $patient_number);
$stmt->bindParam(':oldStartDate', $oldStartDate);
$stmt->bindParam(':oldEndDate', $oldEndDate);
$stmt->bindParam(':serialnum', $oldSerialNum);
$stmt->execute();

if($stmt == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}

$newDevice = $_REQUEST['serialnum'];
$stmt = $connection->prepare("INSERT INTO Wears VALUES (:start_date, :end_date, :patient_number,
:serialnum, :manufacturer)");
$stmt->bindParam(':start_date', $currentTimeDate);
$stmt->bindParam(':end_date', $oldEndDate);
$stmt->bindParam(':patient_number', $patient_number);
$stmt->bindParam(':serialnum', $newDevice);
$stmt->bindParam(':manufacturer', $oldManufacturer);
$stmt->execute();

if($stmt == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}
$connection->commit();
echo("<p>Device was replaced!</p>");
echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");
$connection = null;
}
else
{
    echo("<p>No patient was searched</p>");
    echo("<p><a href=\"getPatient.php\">Search for a patient</a></p>");
}
?>
</body>
</html>

```



Device was replaced!

Turn to the [Home page](#)

Figure 8 – Confirmation of the replacement

IV. Create a new study/new series

The user can also create a new study by filling a form with the patient number, request number, description, date, a doctor and the serial number and manufacturer of the device that will be used (newStudy.php). The web page with the form is represented in Figure 9.

- [newStudy.php:](#)

```
<html>
<body>
<h3><strong><font color= '#66CC00'>New Study</font></strong></h3>
<hr/>
<form action="insertStudy.php" method="post">
<?php
    require 'connectDB.php';
?>
<p>Patient Number:
<select name="patient_number">
<?php
    $sql = "SELECT * FROM Patient";
    $result = $connection->query($sql);

    if($result == FALSE)
    {
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    foreach($result as $row)
    {
        $patient_number = $row['patient_number'];
        $patient_name = $row['name'];
        echo("<option value=\"{$patient_number}\">{$patient_number} | {$patient_name}</option>");
    }
?>
</select>
</p>
<p>Request Number: <input type="text" name="request_number" required/></p>
<p>Description: <input type="text" name="description" required/></p>
<p>Study Date: <input type="text" name="study_date" required/></p>
<p>Doctor:
<select name="doctor_id">
<?php
    $stmt = $connection->prepare("SELECT doctor_id FROM Doctor;");
    $stmt->execute();

    if($stmt == FALSE)
    {
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    foreach($stmt as $row)
    {
        $doctor_id = $row['doctor_id'];
        echo("<option value=\"{$doctor_id}\">{$doctor_id}</option>");
    }
?>
</select>
</p>
```

```

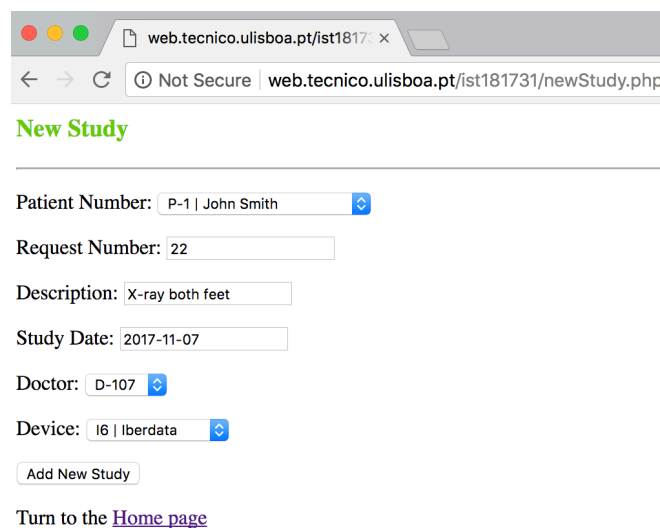
        <p>Device:
        <select name="serialnum">
<?php
    $stmt = $connection->prepare("SELECT serialnum, manufacturer FROM Device WHERE (serialnum NOT IN(SELECT
serialnum FROM Wears WHERE patient_number <> :patient_number)) OR (serialnum IN (SELECT serialnum FROM Wears
WHERE patient_number = :patient_number))");
    $stmt->bindParam(':patient_number', $patient_number);
    $stmt->execute();

    if($stmt == FALSE)
    {
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    foreach($stmt as $row)
    {
        $serialnum = $row['serialnum'];
        $manufacturer = $row['manufacturer'];

        echo("<option value=\"\$serialnum\">\$serialnum | \$manufacturer</option>");
    }
?>
    </select>
    </p>
    <p><input type="submit" value="Add New Study"/></p>
<?php
    echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");
    $connection = null;
?>
</form>
</body>
</html>

```



The screenshot shows a web browser window with the URL `web.tecnico.ulisboa.pt/ist181731/newStudy.php`. The page title is "New Study". The form contains the following elements:

- Patient Number:** A dropdown menu showing "P-1 | John Smith".
- Request Number:** A text input field containing the value "22".
- Description:** A text input field containing the value "X-ray both feet".
- Study Date:** A text input field containing the value "2017-11-07".
- Doctor:** A dropdown menu showing "D-107".
- Device:** A dropdown menu showing "I6 | Iberdata".
- Buttons:** An "Add New Study" button and a link "Turn to the [Home page](#)".

Figure 9 – Add a new study

When the study is created, it's also created a series (insertStudy.php). First, it's verified if the request number already exists in the database, once it has to exist before creating a study. Then, it's performed a query to check what was the doctor that made the request, once the doctor who made the request can't performed the study. After that, the new study is introduced in the database. Before committing the data, some verifications are done such as, checking if: (i) the date of the request precedes the study date; (ii) the request number exists; (iii) the doctor chosen to perform the study isn't the same that made the request; (iv) the patient of the request is the same as the patient of the study.

These tasks are performed in a single transaction. If some of the errors occur, then it's called the `rollback()` function. Otherwise, the changes will be committed. As it can be seen from Figure 10, in both cases a message will be displayed, in the first case Study not created, otherwise Study created.

- [insertStudy.php](#):

```
<html>
<body>
<h3><strong><font color= '#66CC00'>New Study</font></strong></h3>
<hr/>
<?php
    require 'connectDB.php';
    $patient_number = $_REQUEST['patient_number'];
    $request_number = $_REQUEST['request_number'];
    $description = $_REQUEST['description'];
    $study_date = $_REQUEST['study_date'];
    $doctor_id = $_REQUEST['doctor_id'];
    $serialnum = $_REQUEST['serialnum'];
    $series_name = $description;

    if(isset($_REQUEST['patient_number']) && isset($_REQUEST['request_number']) &&
isset($_REQUEST['description']) && isset($_REQUEST['study_date']) && isset($_REQUEST['doctor_id']) &&
isset($_REQUEST['serialnum']))
    {
        if($_REQUEST['study_date'] > date("Y-m-d"))
        {
            echo("<p>Please enter a valid date</p>");
            echo("<p>Study not created</p>");
            echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
            exit();
        }
        /* Begins transaction */
        $connection->beginTransaction();

        /* Verifies if the resquest number exists */
        $sql = "SELECT * FROM Request";
        $result = $connection->query($sql);

        if($result == FALSE)
        {
            $connection->rollback();
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }

        $requestExists = 0;

        foreach($result as $row)
        {
            $request_id = $row['request_number'];
            if($request_id == $request_number)
            {
                $requestExists = 1;
                /* Date of the request has to be previous than the study date */
            }
        }
    }
}
```

```

        $request_date = $row['request_date'];
        $patientOfRequest = $row['patient_number'];
    }
}

/* Checks if the doctor that made the request isn't the same that is going to perform the study */
$doctor = "SELECT doctor_id FROM Request WHERE request_number = $request_number";
$result = $connection->query($doctor);

if($result == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}

foreach($result as $row)
{
    $requestDoctor = $row['doctor_id'];

/* Checks what is the manufacturer of the device with that serial number */
$getManufacturer = "SELECT manufacturer FROM Device WHERE serialnum = '$serialnum'";
$result = $connection->query($getManufacturer);

if($result == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
}

foreach($result as $row)
{
    $manufacturer = $row['manufacturer'];

/* Add new study */
$stmt = $connection->prepare("INSERT INTO Study VALUES (:request_number, :description,
:study_date, :doctor_id, :manufacturer, :serialnum)");
$stmt->bindParam(':request_number', $request_number);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':study_date', $study_date);
$stmt->bindParam(':doctor_id', $doctor_id);
$stmt->bindParam(':manufacturer', $manufacturer);
$stmt->bindParam(':serialnum', $serialnum);

$stmt->execute();

if($stmt == FALSE)
{
    $connection->rollback();
    $info = $connection->errorInfo();

```



```

        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    /* Checks what is the last number of the series added */
    $getSeries = "SELECT series_id FROM Series";
    $result = $connection->query($getSeries);

    if($result == FALSE)
    {
        $connection->rollback();
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    $series_id = ($result->rowCount()) + 1;
    $base_url = "http://web.tecnico.ulisboa.pt/ist181731/series/" . $series_id;

    /* Add new series */
    $stmt = $connection->prepare("INSERT INTO Series VALUES (:series_id, :series_name, :base_url,
:request_number, :description)");
    $stmt->bindParam(':series_id', $series_id);
    $stmt->bindParam(':series_name', $series_name);
    $stmt->bindParam(':base_url', $base_url);
    $stmt->bindParam(':request_number', $request_number);
    $stmt->bindParam(':description', $description);
    $stmt->execute();

    if($stmt == FALSE)
    {
        $connection->rollback();
        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    /* Study date has to be after request date and the request number should exists */
    if($study_date < $request_date)
    {
        $connection->rollback();
        echo("<p>The request date has to precede the study date!</p>");
        echo("<p>Study not created</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else if($requestExists == 0) /* Roll back if the request doesn't exist */
    {
        $connection->rollback();
        echo("<p>The request number doesn't exist!</p>");
        echo("<p>Study not created</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else if($requestDoctor == $doctor_id) /* Roll back if the doctor who made the request is the same as the study */
    {
        $connection->rollback();
        echo("<p>The doctor who makes the request can't make studies about it!</p>");
    }

```

```

        echo("<p>Study not created</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else if($patientOfRequest != $patient_number)
    {
        $connection->rollback();
        echo("<p>The request number doesn't belong to patient number $patient_number!</p>");
        echo("<p>Study not created</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else
    {
        $connection->commit();
        echo("<p>Study created</p>");
        echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");
    }
    $connection = null;
}
else
{
    echo("<p>No form was filled!</p>");
    echo("<p><a href=\"newStudy.php\">Create a new study</a></p>");
}
?>
</body>
</html>

```



Figure 10 – Study created

If the study wasn't created, it could be one of the four reasons described above:

(i) The date of the request has to precede the study date (Figure 11)

The screenshot shows the 'New Study' form in a web browser. The form includes the following fields and controls:

- Patient Number: P-2 | Roger Smith (dropdown menu)
- Request Number: 2 (text input)
- Description: X-ray right hand (text input)
- Study Date: 2017-06-05 (text input)
- Doctor: D-105 (dropdown menu)
- Device: B4 | Bosch (dropdown menu)
- Add New Study (button)

Below the form, there is a link to 'Home page'.

The screenshot shows the 'New Study' page with an error message: 'The request date has to precede the study date!'. Below the error message, it says 'Study not created' and 'Turn to the previous page'.

Figure 11 – Error (i), request date has to precede the study date

(ii) The request number doesn't exist (Figure 12)

web.tecnico.ulisboa.pt

web.tecnico.ulisboa.pt/ist181731/newStudy.php

New Study

Patient Number: P-3 | James Bond

Request Number: 30

Description: X-ray right hand

Study Date: 2017-11-07

Doctor: D-108

Device: M1 | Medtronic

Add New Study

Turn to the [Home page](#)

web.tecnico.ulisboa.pt/ist181731/insertStudy.php

New Study

The request number doesn't exist!

Study not created

Turn to the [previous page](#)

Figure 12 – Error (ii), request number doesn't exist

(iii) The doctor chosen to perform the study is the same that made the request (Figure 13)

web.tecnico.ulisboa.pt

web.tecnico.ulisboa.pt/ist181731/newStudy.php

New Study

Patient Number: P-1 | John Smith

Request Number: 1

Description: X-ray right hand

Study Date: 2017-11-01

Doctor: D-101

Device: B4 | Bosch

Add New Study

Turn to the [Home page](#)

web.tecnico.ulisboa.pt/ist181731/insertStudy.php

New Study

The doctor who makes the request can't make studies about it!

Study not created

Turn to the [previous page](#)

Figure 13 – Error (iii), doctor who requested can't make the study

(iv) The patient of the request isn't the same as the patient of the study

web.tecnico.ulisboa.pt

web.tecnico.ulisboa.pt/ist181731/newStudy.php

New Study

Patient Number: P-10 | Wellington Ramirez

Request Number: 1

Description: X-ray hands

Study Date: 2017-11-07

Doctor: D-104

Device: I4 | Iberdata

Add New Study

Turn to the [Home page](#)

web.tecnico.ulisboa.pt/ist181731/insertStudy.php

New Study

The request number doesn't belong to patient number P-10!

Study not created

Turn to the [previous page](#)

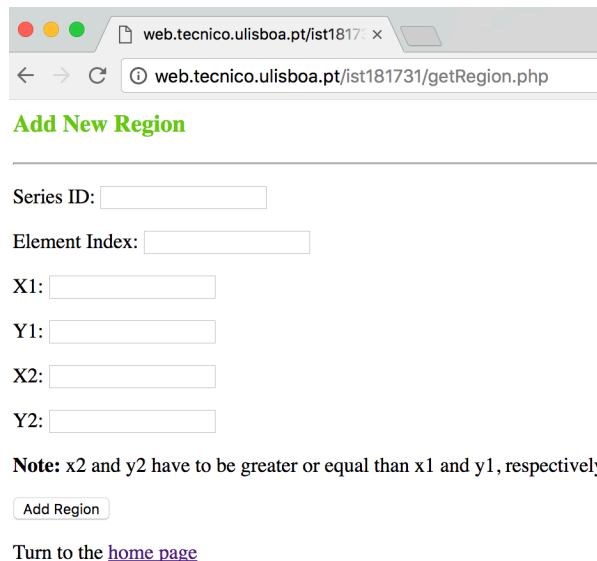
Figure 14 – Error (iv), request number doesn't belong to the given patient

V. Add new region

The last task to perform is to add a new region to a certain element of a series (getRegion.php). To do that, the user has to fill a form, as represented in Figure 15.

- [getRegion.php](#):

```
<html>
  <body>
    <h3><strong><font color= '#66CC00'>Add New Region</font></strong></h3>
    <hr/>
    <form action="newRegion.php" method="post">
      <p>Series ID: <input type="text" name="series_id" required/></p>
      <p>Element Index: <input type="text" name="elem_index" required/></p>
      <p>X1: <input type="text" name="x1" required/></p>
      <p>Y1: <input type="text" name="y1" required/></p>
      <p>X2: <input type="text" name="x2" required/></p>
      <p>Y2: <input type="text" name="y2" required/></p>
      <?php
        echo("<p><strong>Note:</strong> x2 and y2 have to be greater or equal than x1 and y1, respectively</p>");
      ?>
      <p><input type="submit" value="Add Region"/></p>
    </form>
    <?php
      echo("<p> Turn to the <a href='\"homePage.php\"'>home page</a>");
    ?>
  </body>
</html>
```



The screenshot shows a web browser window with the address bar displaying 'web.tecnico.ulisboa.pt/ist181731/getRegion.php'. The page title is 'Add New Region'. The form contains the following elements:

- Series ID:
- Element Index:
- X1:
- Y1:
- X2:
- Y2:
- Note: x2 and y2 have to be greater or equal than x1 and y1, respectively
- Add Region button
- Turn to the [home page](#)

Figure 15 – Form to add a new region

The new record is added to the table Region in newRegion.php file. After inserting that new record in the database, it has to be done a verification that needs the last study of the patient (if there was one). If the description of the last study was the same and this new region doesn't overlap with any region of any element of the last study, then it will be displayed a message that warns to a new clinical evidence for that patient (Figure 16). Otherwise, the region is inserted without warning (Figure 17). Before executing the commit, some validations must be done, such as: (i) check if the series exists and (ii) check if x1 and y1 are equal or greater than x2 and y2, respectively.

- [newRegion.php:](#)

```
<html>
<body>
<h3><strong><font color= '#66CC00'>New Region</font></strong></h3>
<hr/>
<?php
    require 'connectDB.php';

    $series_id = $_REQUEST['series_id'];
    $elem_index = $_REQUEST['elem_index'];
    $x1 = $_REQUEST['x1'];
    $y1 = $_REQUEST['y1'];
    $x2 = $_REQUEST['x2'];
    $y2 = $_REQUEST['y2'];

    if(isset($_REQUEST['series_id']) && isset($_REQUEST['elem_index']) && isset($_REQUEST['x1']) &&
isset($_REQUEST['y1']) && isset($_REQUEST['x2']) && isset($_REQUEST['y2']))
    {
        if($_REQUEST['series_id'] <= 0 || $_REQUEST['elem_index'] <= 0 || $_REQUEST['x1'] < 0 ||
$_REQUEST['y1'] < 0 || $_REQUEST['x2'] < 0 || $_REQUEST['y2'] < 0 || $_REQUEST['x1'] > 1 || $_REQUEST['y1'] > 1 ||
$_REQUEST['x2'] > 1 || $_REQUEST['y2'] > 1 )
        {
            echo("<p>Please enter valid inputs. All inputs have to be greater or equal to zero</p>");
            echo("<p>Turn to the <a href='\"getRegion.php\"'>previous page</a></p>");
            exit();
        }

        /* Begins transaction */
        $connection->beginTransaction();

        $stmt = $connection->prepare("SELECT * FROM Element WHERE series_id = $series_id AND elem_index =
:elem_index");
        $stmt->bindParam(':elem_index', $elem_index);
        $stmt->execute();

        if($stmt == FALSE)
        {
            $connection->rollback();
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }

        $seriesExists = $stmt->rowCount();

        /* Add new region */
        $stmt = $connection->prepare("INSERT INTO Region VALUES (:series_id, :elem_index, :x1, :y1, :x2, :y2)");
        $stmt->bindParam(':series_id', $series_id);
        $stmt->bindParam(':elem_index', $elem_index);
        $stmt->bindParam(':x1', $x1);
        $stmt->bindParam(':y1', $y1);
        $stmt->bindParam(':x2', $x2);
        $stmt->bindParam(':y2', $y2);
        $stmt->execute();

        if($stmt == FALSE)
        {
            $connection->rollback();
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }
    }
}
```

```

    }

    /* Series ID has to exist in the database */
    if($seriesExists == 0)
    {
        $connection->rollback();
        echo("<p>The series or/and the element index don't exist</p>");
        echo("<p>New Region not added</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else if($x1 > $x2 || $y1 > $y2)
    {
        $connection->rollback();
        echo("<p>x2 and y2 have to be greater or equal than x1 and y1, respectively</p>");
        echo("<p>New Region not added</p>");
        echo("<p>Turn to the <a href=\"newStudy.php\">previous page</a></p>");
    }
    else
    {
        $connection->commit();
        echo("<p>New Region Added</p>");

        /* Check the description of the study of the region that was added */
        $stmt = $connection->prepare("SELECT * FROM Study NATURAL JOIN Series WHERE series_id
= :series_id");

        $stmt->bindParam(":series_id", $series_id);
        $stmt->execute();

        if($stmt == FALSE)
        {
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }

        foreach($stmt as $row)
        {
            $newestDesc = $row['description'];
            $newestDate = $row['study_date'];
            $newestRequest = $row['request_number'];
        }

        /* Last study of the patient */
        $stmt = $connection->prepare("SELECT MAX(study_date) AS study_date FROM Study WHERE
study_date IN (SELECT study_date FROM Study WHERE request_number IN (SELECT request_number FROM Request
WHERE patient_number IN (SELECT patient_number FROM Series NATURAL JOIN Request WHERE series_id = :series_id))
AND request_number <> :newestRequest);");
        $stmt->bindParam(":series_id", $series_id);
        $stmt->bindParam(":newestRequest", $newestRequest);
        $stmt->execute();

        if($stmt == FALSE)
        {
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }

        $numberOfRows = $stmt->rowCount();

        /* If there is any previous study of the patient */

```

```

if($numberOfRows == 1)
{
    foreach($stmt as $row)
    {
        $lastDate = $row['study_date'];
    }

    /* If the last study date precedes the study date where the new region is being inserted then the regions will be compared */
    if($lastDate < $newestDate)
    {
        $stmt = $connection->prepare("SELECT * FROM Study NATURAL JOIN
Series NATURAL JOIN Element WHERE study_date = :lastDate AND description = :newestDesc");
        $stmt->bindParam(":lastDate", $lastDate);
        $stmt->bindParam(":newestDesc", $newestDesc);
        $stmt->execute();

        $nrows = $stmt->rowCount();

        /* Means that the last study has the same description */
        if($nrows > 0)
        {
            foreach($stmt as $row)
            {
                $lastSeries = $row['series_id'];
                $lastElements[] = $row['elem_index'];
            }

            /* Checks if any region of any element of the last study overlaps with the new region */
            for ($i = 0; $i < count($lastElements); $i++)
            {
                $lastElemIndex = $lastElements[$i];
                $stmt = $connection->prepare("SELECT region_overlaps_element(:lastSeries,
:lastElemIndex, :x1, :y1, :x2, :y2) AS overlaps");
                $stmt->bindParam(":lastSeries", $lastSeries);
                $stmt->bindParam(":lastElemIndex", $lastElemIndex);
                $stmt->bindParam(":x1", $x1);
                $stmt->bindParam(":y1", $y1);
                $stmt->bindParam(":x2", $x2);
                $stmt->bindParam(":y2", $y2);
                $stmt->execute();

                if($stmt == FALSE)
                {
                    $info = $connection->errorInfo();
                    echo("<p>Error: {$info[2]}</p>");
                    exit();
                }

                foreach($stmt as $row)
                {
                    $overlaps = $row['overlaps'];
                }

                /* If any region of a certain element of the last study doesn't overlap with the new region inserted then throws an alert */
                if($overlaps == 0)
                {
                    $stmt = $connection->prepare("SELECT name, patient_number FROM
Request NATURAL JOIN Series NATURAL JOIN Patient WHERE series_id = :series_id");
                    $stmt->bindParam(":series_id", $series_id);
                    $stmt->execute();

                    if($stmt == FALSE)
                    {

```

```

        $info = $connection->errorInfo();
        echo("<p>Error: {$info[2]}</p>");
        exit();
    }

    foreach($stmt as $row)
    {
        $patient_name = $row['name'];
        $patient_number = $row['patient_number'];

        echo("<p>New clinic evidence for patient number $patient_number,
        $patient_name!</p>");

        break;
    }
}

echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");
}
}
else
{
    echo("<p>The form has to be filled to add a new region</p>");
    echo("<p><a href=\"getRegion.php\">Add a new Region</a></p>");
}
}
$connection = null;
?>
</body>
</html>

```

Add New Region

Series ID:

Element Index:

X1:

Y1:

X2:

Y2:

Note: x2 and y2 have to be greater or equal than x1 and y1, respectively

Turn to the [home page](#)

New Region

New Region Added

New clinic evidence for patient number P-1, John Smith!

Turn to the [Home page](#)

Figure 16 – New region added and displays new evidence

Add New Region

Series ID:

Element Index:

X1:

Y1:

X2:

Y2:

Note: x2 and y2 have to be greater or equal than x1 and y1, respectively

Turn to the [home page](#)

New Region

New Region Added

Turn to the [Home page](#)

Figure 17 – New region added without new evidence

If the new region wasn't added, it could be one of the two reasons described above:

(i) Series doesn't exist (Figure 18)

web.tecnico.ulisboa.pt/ist181731/getRegion

Add New Region

Series ID:

Element Index:

X1:

Y1:

X2:

Y2:

Note: x2 and y2 have to be greater or equal than x1 and y1, respectively

Turn to the [home page](#)

web.tecnico.ulisboa.pt/ist181731/newRegion.php

New Region

The series or/and the element index don't exist

New Region not added

Turn to the [previous page](#)

Figure 18 – Try adding a region to a non-existing series

(ii) x1 and y1 greater than x2 and y2, respectively (Figure 19)

web.tecnico.ulisboa.pt/ist181731/getRegion.php

Add New Region

Series ID:

Element Index:

X1:

Y1:

X2:

Y2:

Note: x2 and y2 have to be greater or equal than x1 and y1, respectively

Turn to the [home page](#)

web.tecnico.ulisboa.pt/ist181731/newRegion.php

New Region

x2 and y2 have to be greater or equal than x1 and y1, respectively

New Region not added

Turn to the [previous page](#)

Figure 19 – Try adding a region with x1 and y1 greater than x2 and y2, respectively