

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

1. **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">

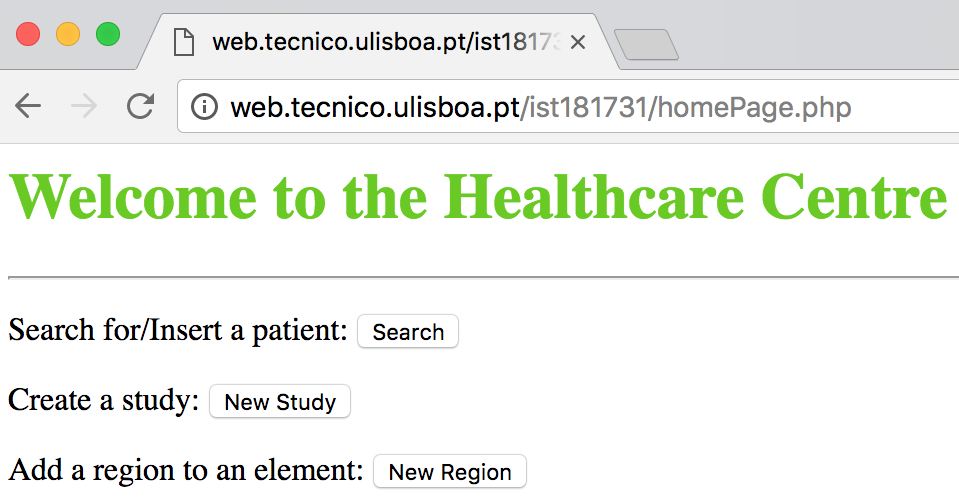
<p>Create a study: <input type="submit" value="New Study"/></p>

</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

1. **Check patient existence**

The first task consists of searching for a patient, giving all 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>

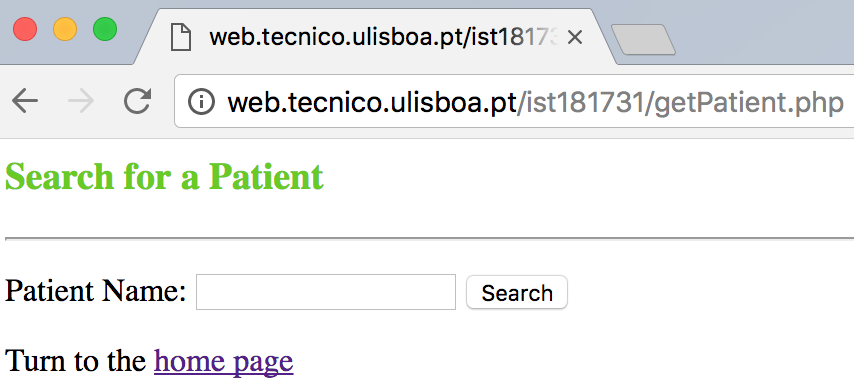
</html>

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 its presented a table with all the results and a possibility of registering a new patient, either there are results or not, once the results given couldn’t match with the patient that is being searched (that is represented 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>

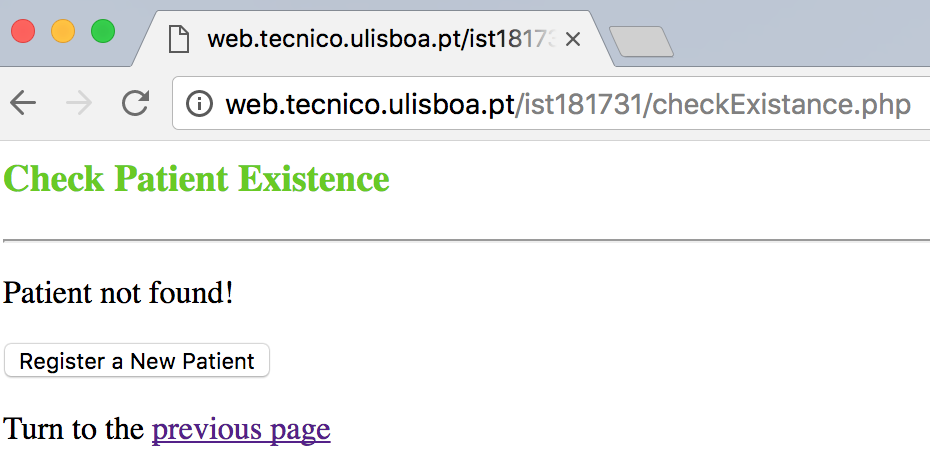
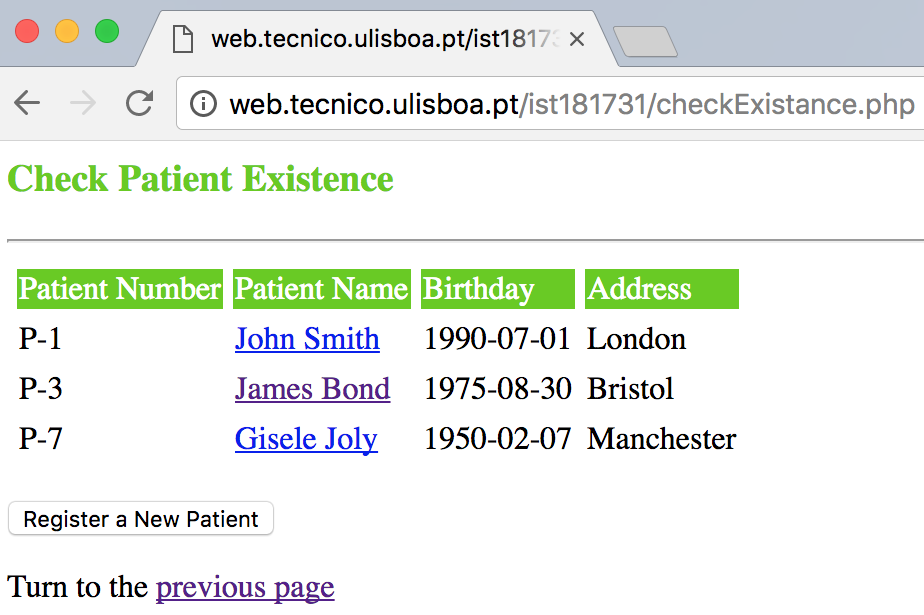


Figure 3 – Display records that contain the substring searched

If the portion of a name doesn’t match with any of the results (if there is one at least), then it’s possible to register a new patient, as well as making another search pressing 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($\_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>");

}

else if(isset($\_REQUEST['newName']) && isset($\_REQUEST['newBirthday']) && isset($\_REQUEST['newAddress']))

{

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>

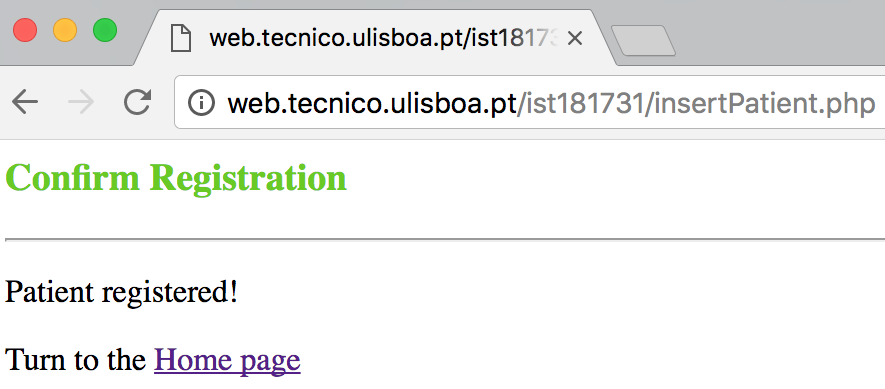


Figure 5 – Confirm registration

1. **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>");

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=\"checkExistance.php?findName=$findName\">previous page</a></p>");

}

else

{

echo("<p>Patient never wore any device!</p>");

echo("<p>Turn to the <a href=\"checkExistance.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>

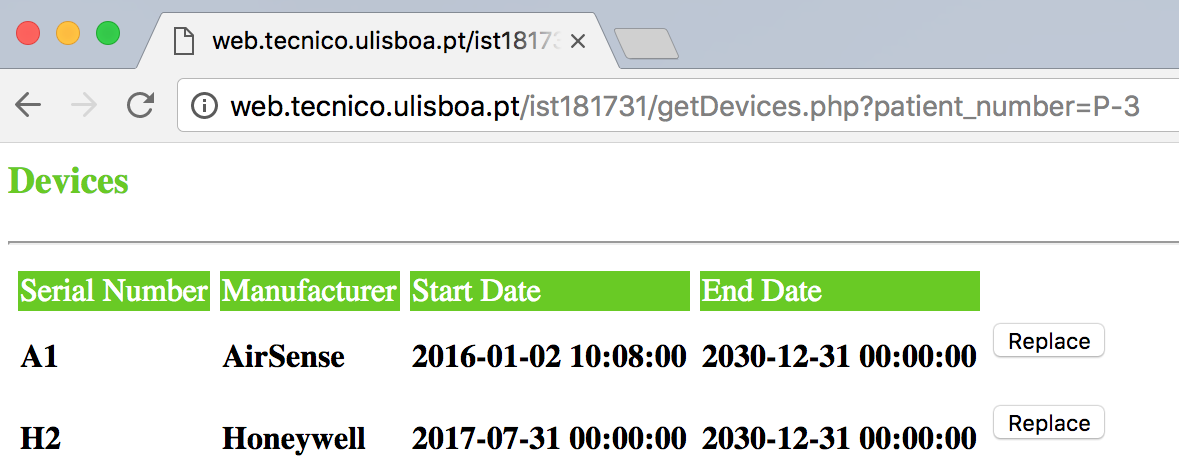


Figure 6 – Devices currently attached to a patient

Each device the patient is wearing at the time 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>

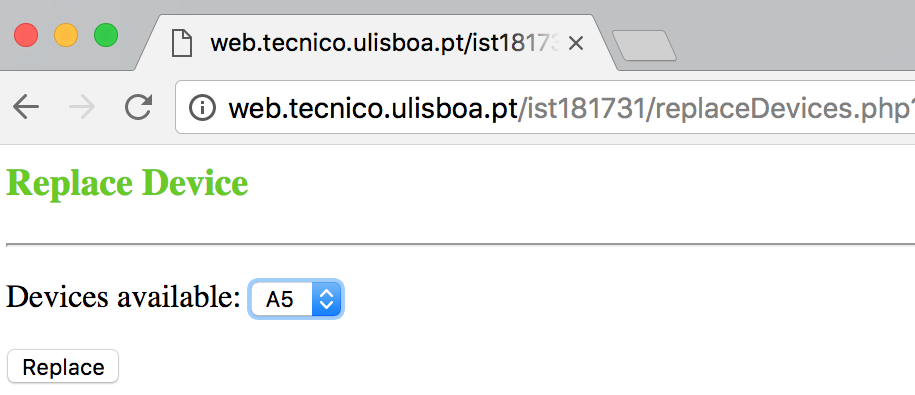
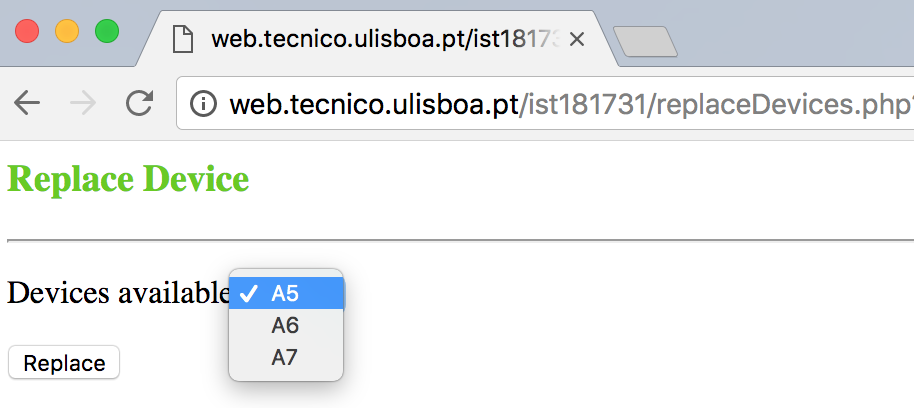


Figure 7 – Devices available to replace

In the insertReplace.php file the first thing that has to be done is to check if the both 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 any error occurs, 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;

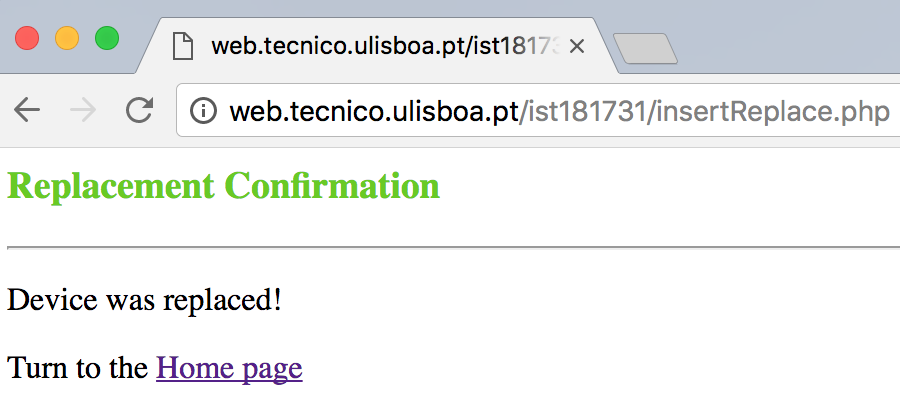
}

else

{

echo("<p>No patient was searched</p>");

echo("<p><a href=\"getPatient.php\">Search for a patient</a></p>");

}

?>

</body>

</html>

Figure 8 – Confirmation of the replacement

1. **Create a new study/new series**

The user can also create a new study 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>

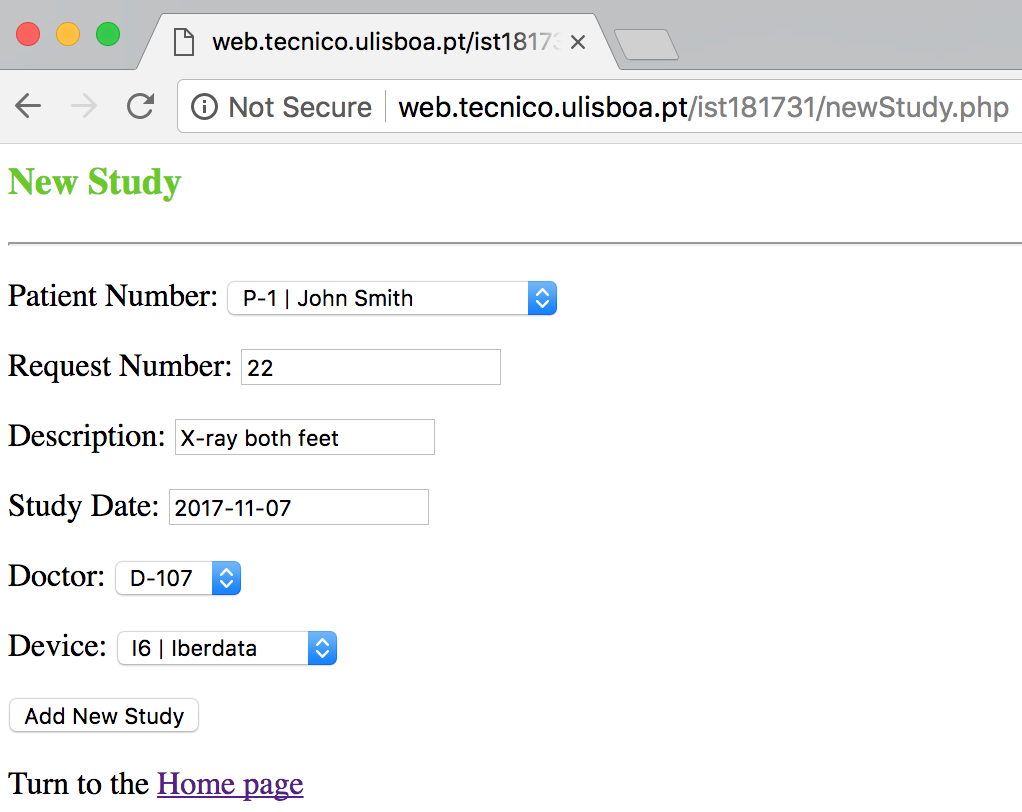
<?php

echo("<p>Turn to the <a href=\"homePage.php\">Home page</a></p>");

$connection = null;

?>

</form>

</body>

</html>

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 it’s introduce 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($\_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>");

}

else if(isset($\_REQUEST['patient\_number']) && isset($\_REQUEST['request\_number']) && isset($\_REQUEST['description']) && isset($\_REQUEST['study\_date']) && isset($\_REQUEST['doctor\_id']) && isset($\_REQUEST['serialnum']))

{

/\* 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:

1. **The date of the request has to precede the study date (Figure 11)**

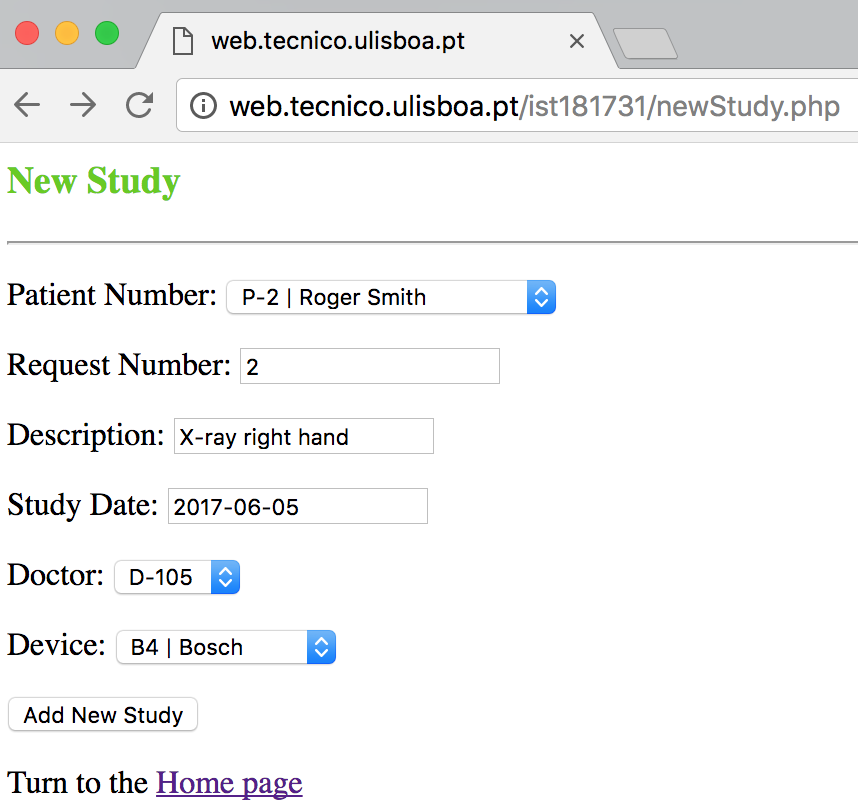
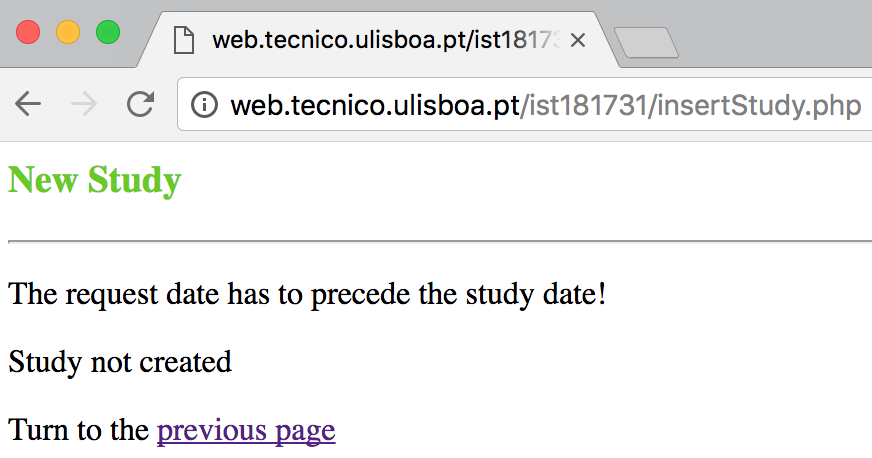


Figure 11 – Error (i), request date has to precede the study date

1. **The request number doesn’t exist (Figure 12)**

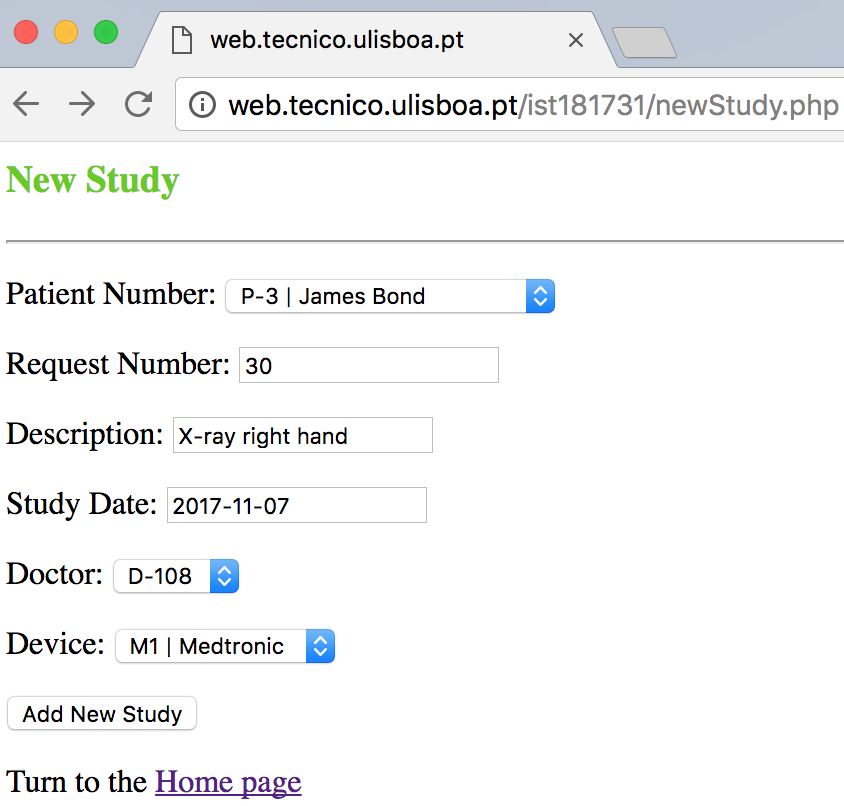


Figure 12 – Error (ii), request number doesn’t exist

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

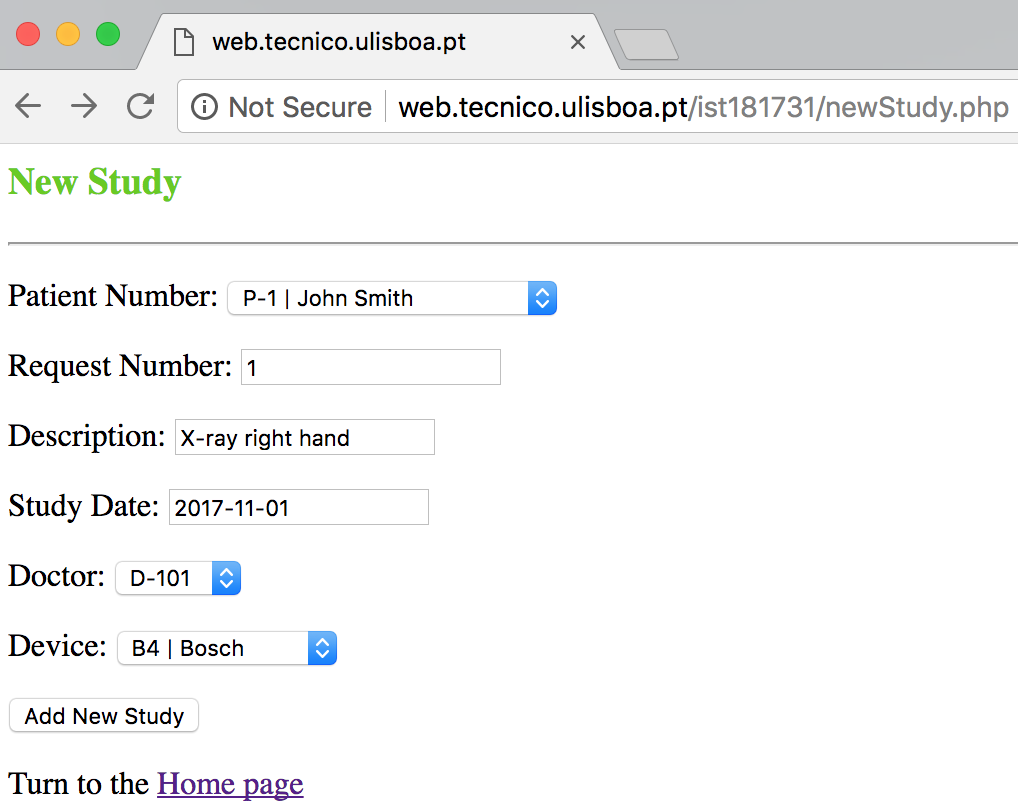


Figure 13 – Error (iii), doctor who requested can’t make the study

1. **The patient of the request isn’t the same as the patient of the study**

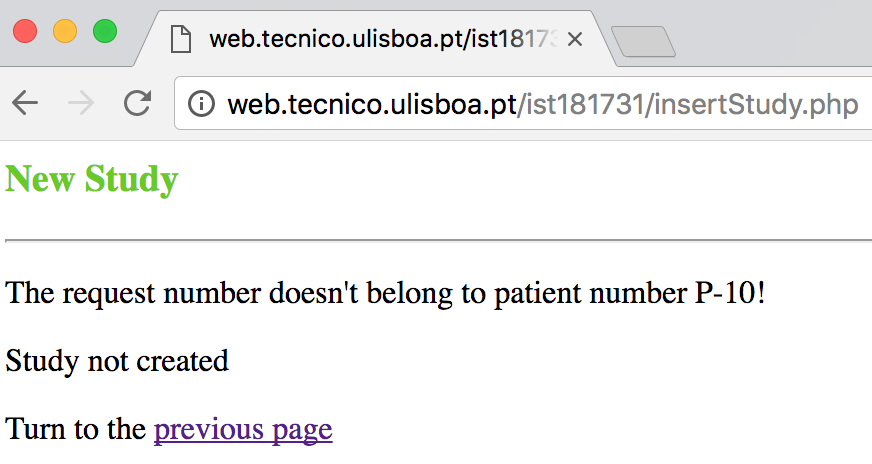
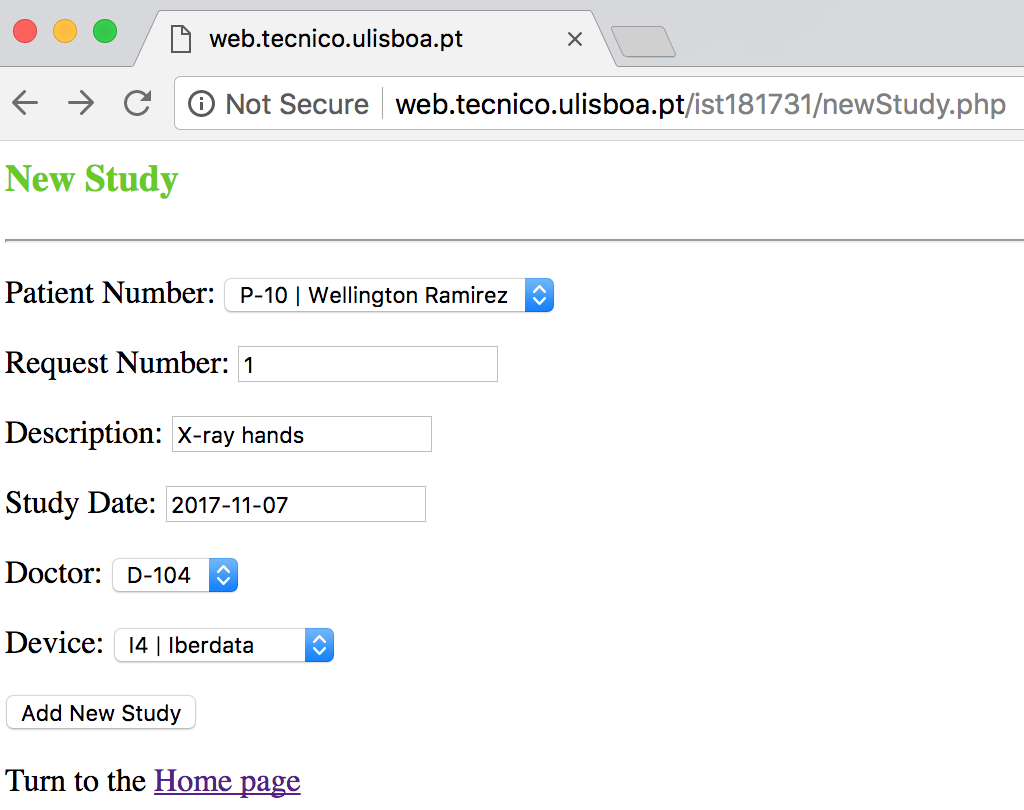


Figure 14 – Error (iv), request number doesn’t belong to the given patient

1. **Add a new region**

The last task to perform is to add a new region to a certain element of a series (getRegion.php). In order 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>Elemente 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 y2, respectively</p>");

?>

<p><input type="submit" value="Add Region"/></p>

</form>

<?php

echo("<p> Turn to the <a href=\"homePage.php\">home page</a>");

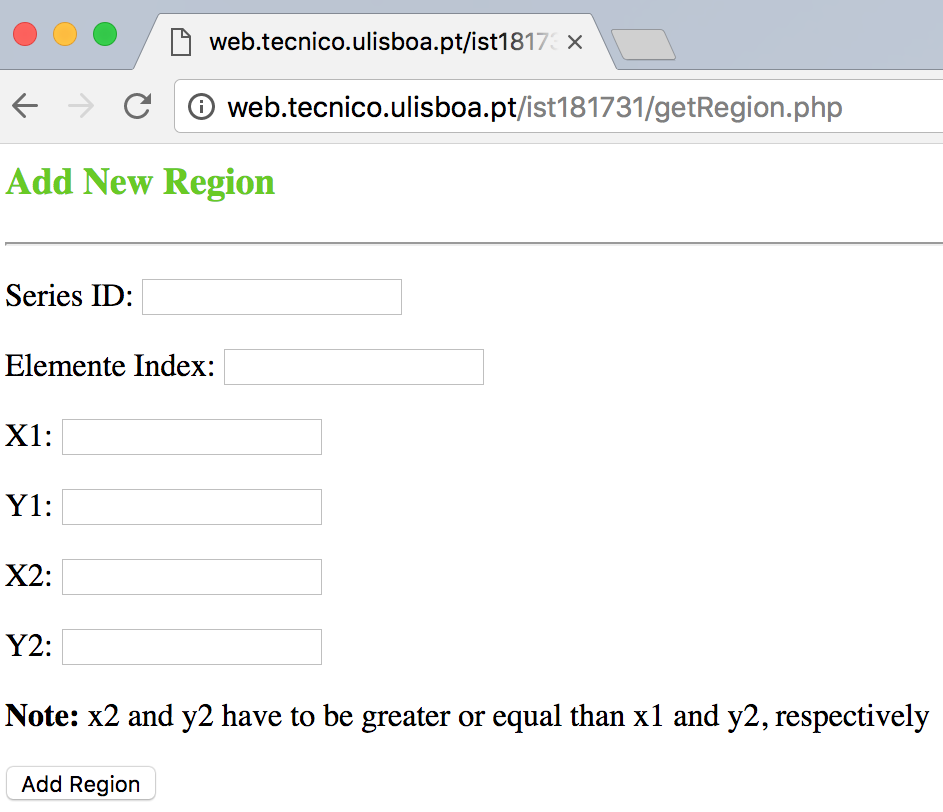
?>

Figure 15 – Form to add a new region

</body>

</html>

The new record is added to the table Region in newRegion.php file. After insert 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 with any warning (Figure 17). Before executing the commit, some validations have to be done, such as: (i) check if the series exists and (ii) check if x1 and y2 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($\_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>");

}

else if(isset($\_REQUEST['series\_id']) && isset($\_REQUEST['elem\_index']) && isset($\_REQUEST['x1']) && isset($\_REQUEST['y1']) && isset($\_REQUEST['x2']) && isset($\_REQUEST['y2']))

{

/\* Begins transaction \*/

$connection->beginTransaction();

$sql = "SELECT \* FROM Element WHERE series\_id = $series\_id AND elem\_index = $elem\_index";

$result = $connection->query($sql);

if($result == FALSE)

{

$connection->rollback();

$info = $connection->errorInfo();

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

exit();

}

$seriesExists = $result->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 y2, 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 \*/

$sql = "SELECT description FROM Series WHERE series\_id = $series\_id";

$result = $connection->query($sql);

if($result == FALSE)

{

$info = $connection->errorInfo();

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

exit();

}

foreach($result as $row)

{

$newestDesc = $row['description'];

}

/\* Check if region does not overlap with any of the regions of an element of the last study of the patient \*/

$sql = "SELECT \* FROM Study NATURAL JOIN Series NATURAL JOIN Element WHERE request\_number IN ((SELECT request\_number FROM Request WHERE patient\_number IN (SELECT patient\_number FROM Request WHERE request\_number IN (SELECT request\_number FROM Series WHERE series\_id = 1 )))) AND study\_date >= all(SELECT study\_date FROM Study)";

$result = $connection->query($sql);

if($result == FALSE)

{

$info = $connection->errorInfo();

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

exit();

}

foreach($result as $row)

{

$lastStudyDesc = $row['description'];

$lastSeries = $row['series\_id'];

$lastElements[] = $row['elem\_index'];

}

/\* If the description of the last study is the same as the description of the series in which the new region was added \*/

if($newestDesc == $lastStudyDesc)

{

for ($i = 0; $i < count($lastElements); $i++)

{

$lastElemmIndex = $lastElements[$i];

$sql = "SELECT region\_overlaps\_element($lastSeries, $lastElemmIndex, $x1, $y1, $x2, $y2) AS overlaps";

$result = $connection->query($sql);

if($result == FALSE)

{

$info = $connection->errorInfo();

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

exit();

}

foreach($result 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)

{

$sql = "SELECT name, patient\_number FROM Request NATURAL JOIN Series NATURAL JOIN Patient WHERE series\_id = $series\_id";

$result = $connection->query($sql);

if($result == FALSE)

{

$info = $connection->errorInfo();

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

exit();

}

foreach($result 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>

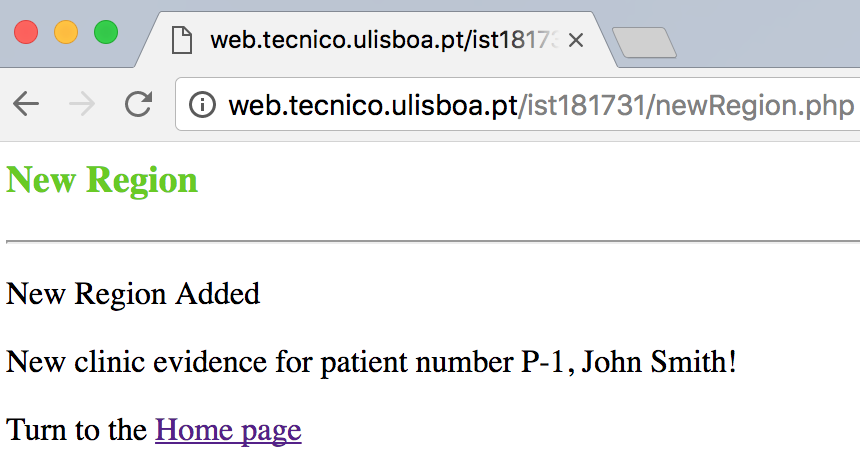
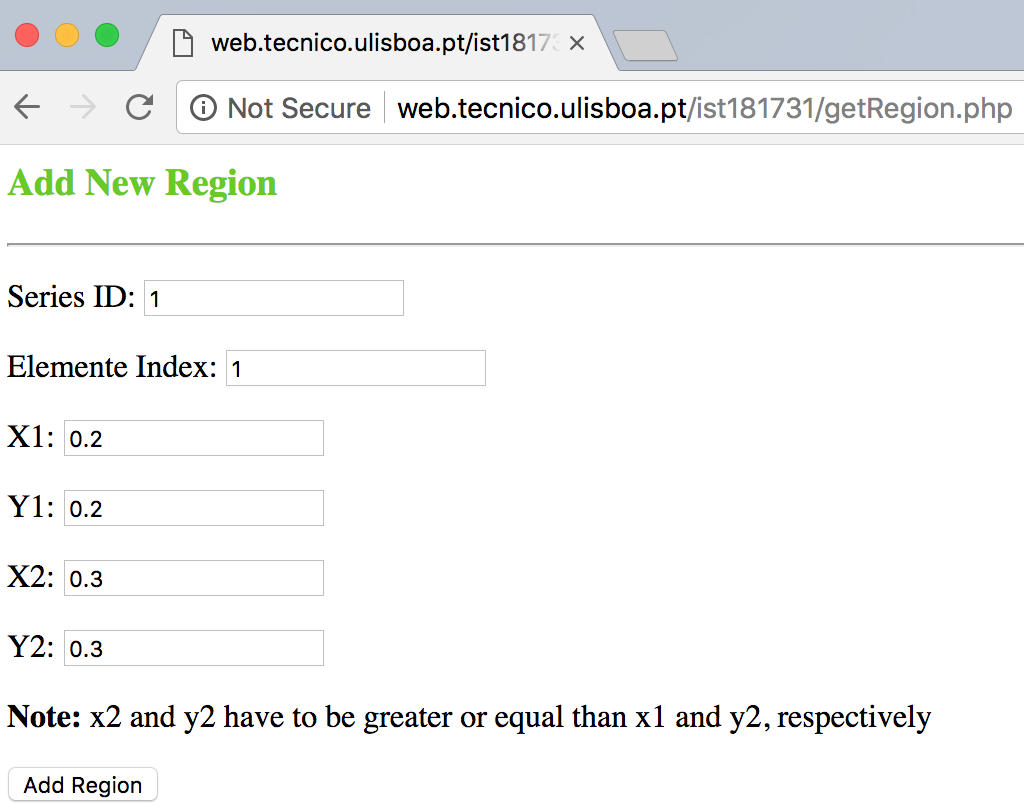


Figure 16 – New region added and displays new evidence

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

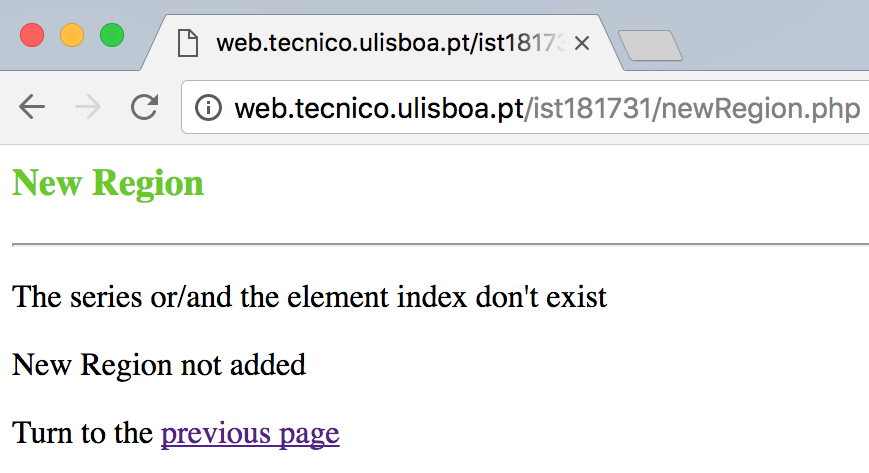
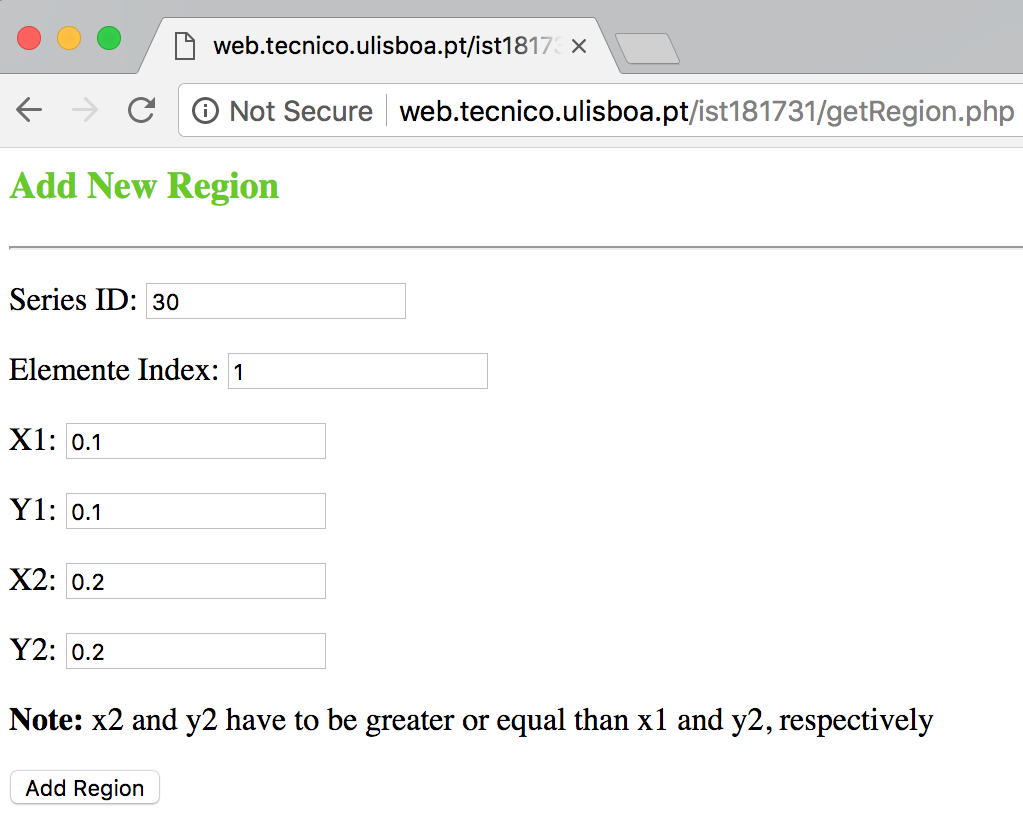
1. **Series doesn’t exist (Figure 18)**

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

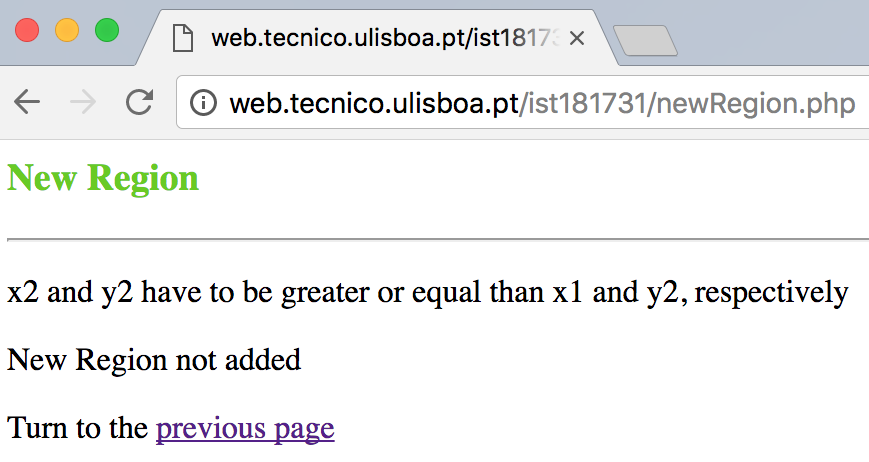
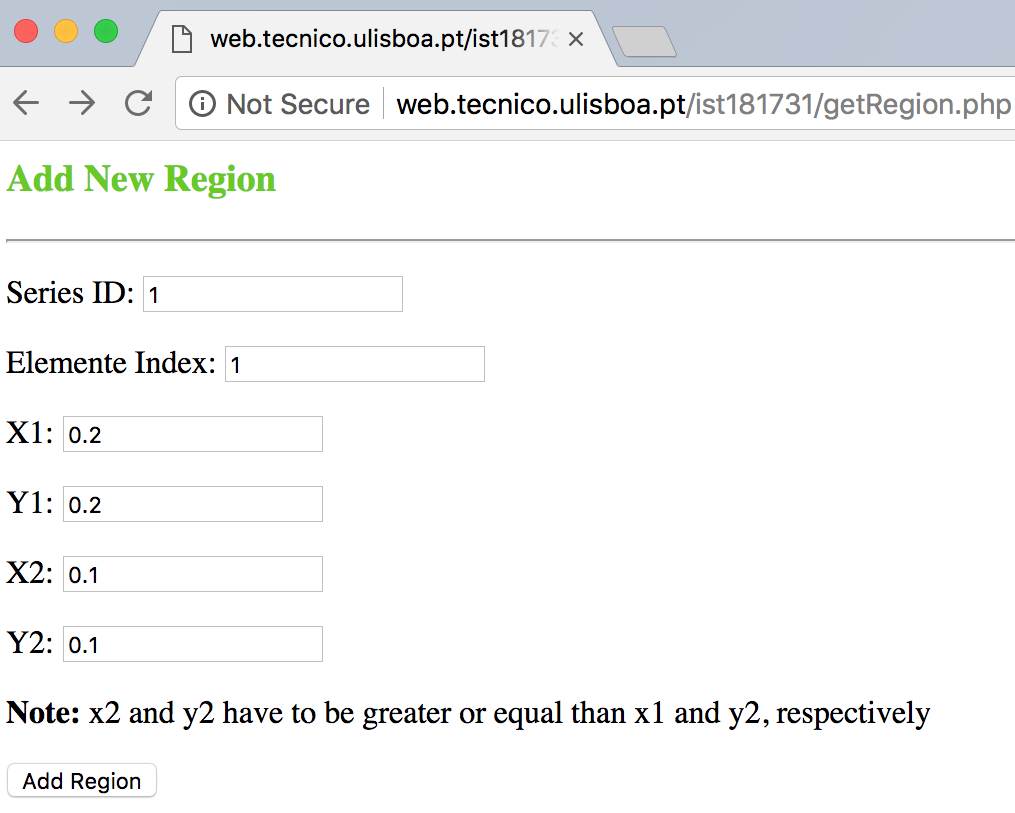
1. **x1 and y1 greater than x2 and y2, respectively (Figure 19)**

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