TÉCNICO LISBOA

Using the Database

Information Systems and Databases

AUTHORS

Jose Silva - 84109
Luís Simões - 81282
Pedro Ferreira - 84166

November 24, 2022

Contents

1 Introduction

This project concerns the development of an information system supporting the day-to-day operations of a dental clinic. This third part of the project concerns the creation of a web application to access the database created on the previous parts of the project. During this project, all interactions with the database that the values are selected by the user are protected against SQL injections through Prepared Statements.

The web application can be found here or go to the link http://web.ist.utl.pt/ist425466/doctorsearch.php.

2 Web Application Using the Database - 1/3

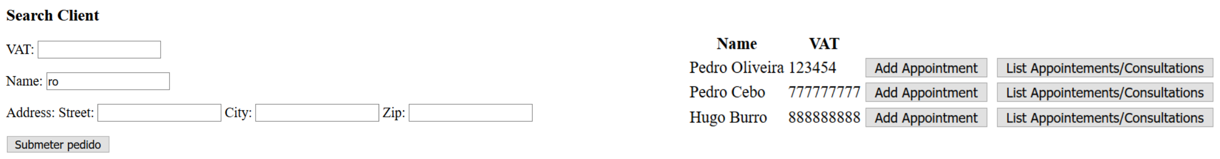
2.1 Search a client

We were asked to search for a client even if just part of the information about him was provided. To do so, it was used the code shown above.

Listing 1: Search client

```
if(!empty($VAT)||!empty($Name)||!empty($Street)||!empty($City)||!empty($Zip))
{
    $sVAT = "%{$_REQUEST['VAT']}%";
    $sName = "%{$_REQUEST['Name']}%";
    $sStreet = "%{$_REQUEST['Street']}%";
    $sCity = "%{$_REQUEST['City']}%";
    $sZip = "%{$_REQUEST['Zip']}%";
    $result = $connection->prepare("SELECT VAT, name FROM client WHERE client.VAT LIKE ? and client.name LIKE ? and client.street LIKE ? and client.city LIKE ? and client.zip LIKE ?");
    $result->execute(array($sVAT, $sName, $sStreet, $sCity, $sZip));



    if ($result->rowCount() == 0)
    {
    echo("<p>There is no client with such info.</p>");
    }
    else
    {
        echo("<table>");
        echo("<tr> <th>Name</th> <th>VAT</th> <th></th> </tr>");
        foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
        {
            echo("<tr><form action='newAppointment.php' method='post'>");
            echo("<td>{$row['name']}</td> <td><input type='hidden' name='VAT' value='{$row['VAT']}'>{$row['VAT']} </td>");
            echo("<td>  <input type='submit' value='Add Appointment'/>  </td>");
26            echo("</form>");
27       echo("<form action='listAppointments.php' method='post'>");
28            echo("<td><input type='hidden' name='VAT' value='{$row['VAT']}'></td>");
29            echo("<td><input type='submit' value='List Appointements/Consultations'/></td>");
30            echo("</form> </tr>");
31
32         }
33         echo("</table>");
34     }
35 }
```

The query won't run if there is no information about the client.



Search Client

VAT:

Name: ro

Address: Street: City: Zip:

Submeter pedido

Name	VAT		
Pedro Oliveira	123454	Add Appointment	List Appointements/Consultations
Pedro Cebo	777777777	Add Appointment	List Appointements/Consultations
Hugo Burro	888888888	Add Appointment	List Appointements/Consultations

(a) inputs

(b) result

Figure 1: Web page - search client

2.2 Add Appointment

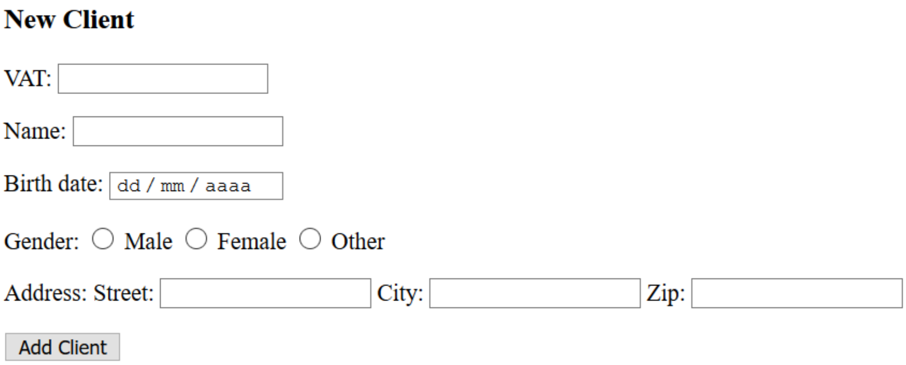
As shown on listing 1 line 25, when the results are printed exists the possibility of registering a new appointment for that client. This submit form will change the window to the *newAppointments.php*, sending the *VAT_client* by post, where the query to insert a new appointment will be done.

2.3 Create a new client

There is also the possibility of insert a new client in the database. The code to do so is shown on the listing below and the web page is shown on fig 2.

Listing 2: New client

```
<form action="newClient.php" method="post">
    <h3> New Client </h3>
    <p>VAT: <input type="text" name="VAT"/></p>
    <p>Name: <input type="text" name="Name"/></p>
    <p>Birth date: <input type="date" name="birth_date"></p>
    <p>Gender: <input type="radio" name="gender" value="M" > Male
        <input type="radio" name="gender" value="F"> Female
        <input type="radio" name="gender" value="O"> Other</p>
    <p>Address: Street: <input type="text" name="Street"/> City: <input type="text" name="City"/> Zip: <input type="text" name="Zip"/></p>

    <input type="submit" value ='Add Client'/>
</form>
```



New Client

VAT:

Name:

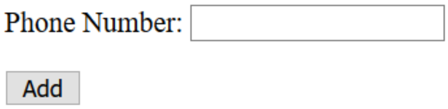
Birth date: dd / mm / aaaa

Gender: ○ Male ○ Female ○ Other

Address: Street: City: Zip:

Figure 2: Web page - new client

When the user presses *Add Client*, he will be redirected to a page where the query that will populate the database will be done. On this page the user can also add phone numbers that will be linked to the new client (figure 3).



Phone Number:

Figure 3: Web page - add phone number

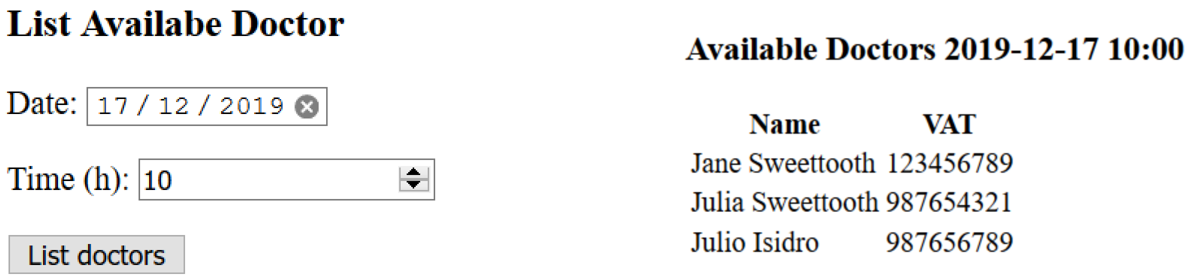
2.4 List available doctors

This function of the program appear two times, it is on the main page and on the *newAppointment.php*.

Listing 3: Check Availability

```
$app_timestamp = $_REQUEST['app_date'].' '.$_REQUEST['app_time'].':00';

$time1 = $_REQUEST['app_time'].':00';
$time2 = "01:00";

$time2 = date("H:i",strtotime($time1)+strtotime($time2));

$app_timestamp2 = $_REQUEST['app_date'].' '.$time2;
if(!empty($_REQUEST['app_date'])&&!empty($_REQUEST['app_time']))
{
    echo("<h3>Available Doctors $app_timestamp</h3>");

    $result = $connection->prepare("SELECT doc.name, doc.VAT from employee doc,
       doctor where doc.VAT = doctor.VAT and doc.VAT not in( select app.VAT_doctor
       from appointment app where app.VAT_doctor and app.date_timestamp BETWEEN ?
       AND ?) group by doc.VAT");
    $result->execute(array($app_timestamp, $app_timestamp2));

...
}
```

On the example above, the date has to be changed to the right format in order be possible to check the time interval on the database. the time slot is a number input because the minutes are irrelevant to the appointment.



List Availabe Doctor

Date: 17 / 12 / 2019

Time (h): 10

(a) inputs

Available Doctors 2019-12-17 10:00

Name	VAT
Jane Sweettooth	123456789
Julia Sweettooth	987654321
Julio Isidro	987656789

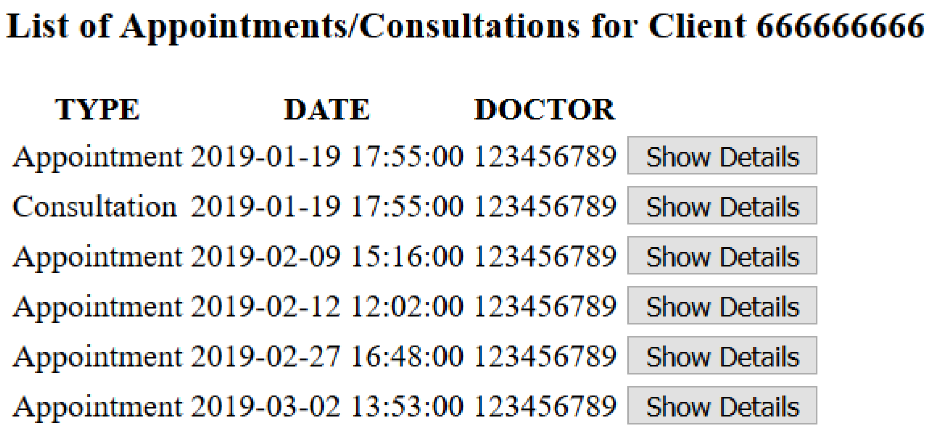
(b) result

Figure 4: Web page - list available doctors

3 Web Application Using the Database - 2/3

3.1 List of Appointments/Consultations

To present a list of appointments/consultations it was simply used a natural join between appointments and consultations (since both have the same primary keys) ordered chronologically.



List of Appointments/Consultations for Client 666666666

TYPE	DATE	DOCTOR	
Appointment	2019-01-19 17:55:00	123456789	Show Details
Consultation	2019-01-19 17:55:00	123456789	Show Details
Appointment	2019-02-09 15:16:00	123456789	Show Details
Appointment	2019-02-12 12:02:00	123456789	Show Details
Appointment	2019-02-27 16:48:00	123456789	Show Details
Appointment	2019-03-02 13:53:00	123456789	Show Details

Figure 5: Web page - List of Appointments/Consultations - *listAppointments.php*

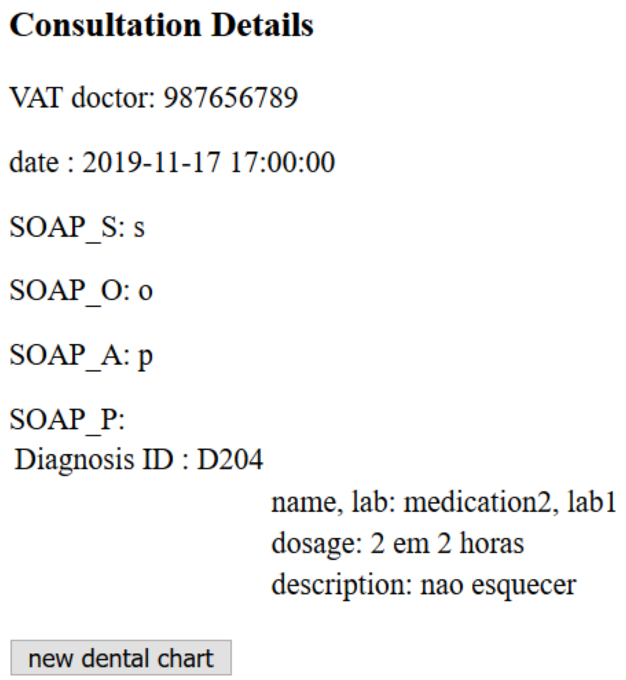
Listing 4: List of Appointments/Consultations (listAppointments.php)

```php
$VAT = $_REQUEST['VAT'];

$result = $connection->prepare("SELECT * FROM appointment NATURAL LEFT OUTER JOIN consultation where VAT_client = ? order by date_timestamp");
$result->execute(array($VAT));

$nrows = $result->rowCount();
if ($nrows == 0)
{
  echo("<p>There are no records for this client.</p>");
}
else
{
  echo("<table>");
  echo("<tr> <th>TYPE</th> <th>DATE</th> <th>DOCTOR</th> </tr>");
  foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
  {
    $date = $row['date_timestamp'];
    $doctor = $row['VAT_doctor'];
    echo("<tr>");
    echo("<td>Appointment</td> <td>{$date}</td> <td>{$doctor}</td>");
    echo("<form action='appointment.php' method='post'>");
    echo("<input type='hidden' name='date' value='{$date}'>");
    echo("<input type='hidden' name='doctor' value='{$doctor}'>");
    echo("<input type='hidden' name='description' value='{$row['description']}'>");
    echo("<input type='hidden' name='client' value='{$VAT}'>");
    echo("<td><input type='submit' value='Show Details'/></td>");
    echo("</form>");
    echo("</tr>");
    if (!is_null($row['SOAP_S']) or !is_null($row['SOAP_O']) or !is_null($row['SOAP_A']) or !is_null($row['SOAP_P']))
    {
      echo("<tr>");
      echo("<td>Consultation</td> <td>{$row['date_timestamp']}</td> <td>{$row['VAT_doctor']}</td>");
      echo("<form action='consultation.php' method='post'>");
      echo("<input type='hidden' name='date' value='{$date}'>");
      echo("<input type='hidden' name='doctor' value='{$doctor}'>");
      echo("<input type='hidden' name='soap_s' value='{$row['SOAP_S']}'>");
      echo("<input type='hidden' name='soap_o' value='{$row['SOAP_O']}'>");
      echo("<input type='hidden' name='soap_a' value='{$row['SOAP_A']}'>");
      echo("<input type='hidden' name='soap_a' value='{$row['SOAP_P']}'>");
      echo("<td><input type='submit' value='Show Details'/></td>");
      echo("</form>");
      echo("</tr>");
    }
  }
  echo("</table>");
}
```

Every appointment and consultation has a *Show Details* button (Figure 5), which will lead the user to a page displaying the information associated to that specific consultation or appointment (Figure 7). All the information is sent as hidden variables so that there is no need to perform a query on the next web page.

3.2 Consultation Information

As far as the page displaying the information regarding an appointment is concerned, it is basically a table showing the appointment's information and a button to insert a new procedure in that consultation. The code of this page is shown below.



Consultation Details

VAT doctor: 987656789

date : 2019-11-17 17:00:00

SOAP_S: s

SOAP_O: o

SOAP_A: p

SOAP_P:
Diagnosis ID : D204
name, lab: medication2, lab1
dosage: 2 em 2 horas
description: nao esquecer

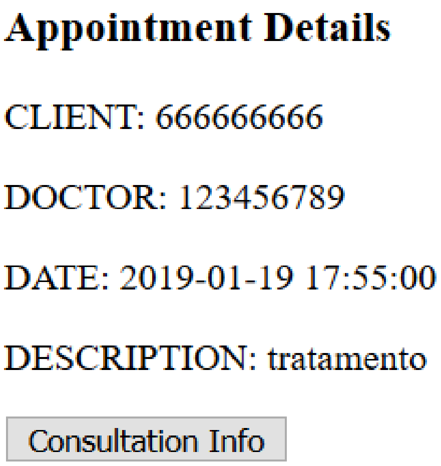
Figure 6: Web page - Consultation Details - *consultation.php*

Listing 5: Consultation Information (consultation.php)

```
$doctor = $_REQUEST['doctor'];
$date = $_REQUEST['date'];



echo("<p>VAT doctor: $doctor");
echo("<p>date : {$_REQUEST['date']}");
echo("<p>SOAP_S: {$_REQUEST['soap_s']}");
echo("<p>SOAP_O: {$_REQUEST['soap_o']}");
echo("<p>SOAP_A: {$_REQUEST['soap_a']}");
echo("<p>SOAP_P: {$_REQUEST['soap_p']}");


$result = $connection->prepare("select * from consultation_diagnostic where VAT_doctor = ? and date_timestamp = ? ");
$result->execute(array($doctor, $date));


$nrows = $result->rowCount();
if ($nrows == 0)
{
echo("<p>There is no Diagnosis.</p>");
}
else
{
    foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
    {
        echo("<table>");
        echo("<tr><td>Diagnosis ID : {$row['ID']}</td></tr>");


        $result2 = $connection->prepare("select * from prescription as p where p.VAT_doctor=? and p.date_timestamp = ? group by p.name");
        $result2->execute(array($doctor, $date));
        $nrows2 = $result->rowCount();
        if($nrows2 == 0)
        {
            echo("<tr><td></td>");
            echo("<td>No prescription</td>");
            echo("</tr>");

        }
        else
        {
            foreach($result2->fetchAll(PDO::FETCH_ASSOC) as $presc)
            {
                echo("<tr><td></td>");
                echo("<td>name, lab: {$presc['name']}, {$presc['lab']} </td>");
                echo("</tr>");
                echo("<tr><td></td>");
                echo("<td> dosage: {$presc['dosage']}</td>");
                echo("</tr>");
                echo("<tr><td></td>");
                echo("<td> description: {$presc['description']}</td>");
                echo("</tr>");
            }
        }

        echo("</table>");
    }
}

echo("<form action='dental_charting.php' method='post'>");
echo("<input type='hidden' name = 'doctor' value = '$doctor'>");
echo("<input type='hidden' name = 'date' value = '$date'>");
```

3.3 Appointment Information

About the page displaying the information about a certain appointment there is no "magic" behind. Nevertheless, it has a button named *Consultation Info* which will send the user to a page where he can create the consultation that is directly related with the displayed appointment, or simply update information about that consultation if it was already created. The code for this page is on Listing 7.



Appointment Details

CLIENT: 666666666

DOCTOR: 123456789

DATE: 2019-01-19 17:55:00

DESCRIPTION: tratamento

Figure 7: Web page - Appointment Details - *appointment.php*

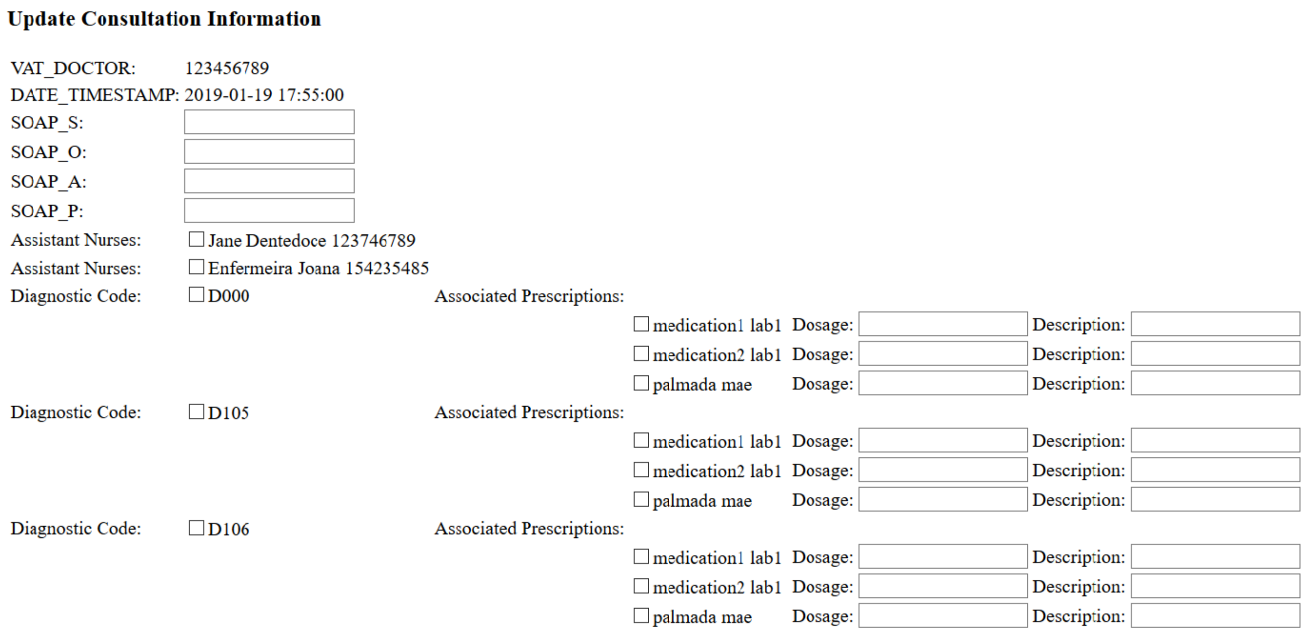
Listing 6: Appointment Information (appointment.php)

```
<html>
    <body>
    <h3>Appointment Details</h3>
    <?php

    $client = $_REQUEST['client'];
    $date = $_REQUEST['date'];
    $doctor = $_REQUEST['doctor'];
    $description = $_REQUEST['description'];

```

```
 echo("<p>CLIENT: {$client}");
 echo("<p>DOCTOR: {$doctor}");
 echo("<p>DATE: {$date}");
 echo("<p>DESCRIPTION: {$description}");
 echo("<p><form action='newConsultation.php' method='post'>");
 echo("<input type='hidden' name='doctor' value='{$doctor}'>");
 echo("<input type='hidden' name='date' value='{$date}'>");
 echo("<input type='submit' value='Consultation Info'/>");
 echo("</form></p>");

 ?>
 </body>
</html>
```

3.4 Creating/Updating a Consultation

On this web page (Figure 8), the user can update whatever information he desires about the corresponding consultation. If this consultation was already created, the corresponding information will already appear on the appropriate spaces, but the user can always update that information.



Update Consultation Information

VAT_DOCTOR: 123456789
DATE_TIMESTAMP: 2019-01-19 17:55:00
SOAP_S:
SOAP_O:
SOAP_A:
SOAP_P:
Assistant Nurses: □ Jane Dentedoce 123746789
Assistant Nurses: □ Enfermeira Joana 154235485
Diagnostic Code: □ D000 Associated Prescriptions:
□ medication1 lab1 Dosage: Description:
□ medication2 lab1 Dosage: Description:
□ palmada mae Dosage: Description:
Diagnostic Code: □ D105 Associated Prescriptions:
□ medication1 lab1 Dosage: Description:
□ medication2 lab1 Dosage: Description:
□ palmada mae Dosage: Description:
Diagnostic Code: □ D106 Associated Prescriptions:
□ medication1 lab1 Dosage: Description:
□ medication2 lab1 Dosage: Description:
□ palmada mae Dosage: Description:

Figure 8: Web page - Creating/Updating a Consultation - *newConsultation.php*

The following code on Listing 7 is divided in two parts, when the consultation was not yet created (line 66 to 114) and when the consultation already existed on the database (line 114 to 227).

Most of the queries are re-utilized for both parts since when we use the *NOT IN* operator, if the consultation does not exist, it would simply list everything as it is supposed.

The first and second parts prepare the information input by the user to send to a page which will do the *INSERTS*, *UPDATES* and *DELETIONS* necessary. Since it is possible to have multiple nurses, prescriptions and medication assigned, the information is stored in arrays, and since the medication depends on the associated prescription, it has to be stored on a multidimensional array, where the ID of the prescription is one of the keys, the same happens for the dosage and prescription which have to be indexed by the prescription ID and by the (name, lab) of the corresponding medication. Note: Since this project assumes that our database can have multiple nurses assigned to the same consultation, we changed the database created on Part2 adding the nurse VAT as a primary key of consultation_assistant.

The difference for the already created consultations is that we wish to present the information that is already on the database, this justifies the use of queries that distinguish nurses, diagnosis and prescriptions that are assigned to this consultations, from the available nurses and all of the diagnosis and prescriptions that can be assign to the consultation. The ones assigned will appear with the "checkbox" already "checked". There's also a need to store arrays corresponding to what was previously on the database, what the user wants to keep and what he wants to add, in order to do the deletions, updates and insertions. Note: Since the prescriptions are different for each diagnosis, an SQL query has to be executed for every already assigned diagnostic.

Also of notice that in order to simplify the indexing of the primary keys of medications (name, lab) we decided to serialize each entry of a medication into a string, which are then stored on an array, and will then be unserialized on the pages responsible for the inserts, updates and deletions.

Listing 7: Creating/Updating a Consultation (newConsultation.php)

```php
$date = $_REQUEST['date'];
$doctor = $_REQUEST['doctor'];

$result = $connection->prepare("SELECT * FROM consultation WHERE VAT_doctor = ? and date_timestamp = ?");
$result->execute(array($doctor, $date));
$error = $result->errorInfo();
if ($error[1] != ''){
  echo("<p>Error: {$error[2]}</p>");
}

$nrows = $result->rowCount();

// Get nurses already assigned to this consultation
$result2 = $connection->prepare("SELECT employee.name AS name, employee.VAT AS VAT FROM consultation_assistant AS ca, employee WHERE ca.VAT_doctor = ? and ca.date_timestamp = ? and ca.VAT_nurse = employee.VAT");
$result2->execute(array($doctor, $date));

$error2 = $result2->errorInfo();
if ($error2[1] != ''){
  echo("<p>Error: {$error2[2]}</p>");
}
$nrows2 = $result2->rowCount();

// Get nurses available for the desired timestamp
$result3 = $connection->prepare("SELECT employee.name AS name, employee.VAT AS VAT FROM nurse, employee WHERE nurse.VAT NOT IN (SELECT VAT_nurse FROM consultation_assistant WHERE date_timestamp = ?) AND nurse.VAT = employee.VAT");
```

```php
$result3->execute(array($date));

$error3 = $result3->errorInfo();
if ($error3[1] != ''){
  echo("<p>Error: {$error3[2]}</p>");
}
$nrows3 = $result3->rowCount();

//Get diagnostic codes already assigned to this consultation
$result_consult_diagnostic = $connection->prepare("SELECT ID FROM consultation_diagnostic WHERE VAT_doctor = ? and date_timestamp = ?");
$result_consult_diagnostic->execute(array($doctor,$date));

$error4 = $result_consult_diagnostic->errorInfo();
if ($error4[1] != ''){
  echo("<p>Error: {$error4[2]}</p>");
}

$nrows_consult_diagnostic = $result_consult_diagnostic->rowCount();

// Get diagnostic codes
$result_diagcodes = $connection->prepare("SELECT ID FROM diagnostic_code WHERE ID NOT IN (SELECT ID FROM consultation_diagnostic WHERE VAT_doctor = ? and date_timestamp = ?)");
$result_diagcodes->execute(array($doctor,$date));
$error5 = $result_diagcodes->errorInfo();
if ($error5[1] != ''){
  echo("<p>Error: {$error5[2]}</p>");
}
$nrows_diagcodes = $result_diagcodes->rowCount();

// Get medication
$sql5 = "SELECT * FROM medication";
$result_medication = $connection->query($sql5);
$nrows_medication = $result_medication->rowCount();
foreach($result_medication as $meds)
{
  $med_array[] = array("name" => $meds['name'], "lab" => $meds['lab']);
}

echo("<table>");
echo("<tr><td>VAT_DOCTOR:</td> <td>{$doctor}</td></tr>");
echo("<tr><td>DATE_TIMESTAMP:</td> <td>{$date}</td></tr>");

// if the consultation was not yet created
if ($nrows == 0)
{
```

```
echo("<form action='addConsultation.php' method='post'>");
echo("<input type='hidden' name='doctor' value='{$doctor}'>");
echo("<input type='hidden' name='date' value='{$date}'>");
echo("<tr><td>SOAP_S:</td> <td><input type='text' name='SOAP_S'/></td></tr>");
echo("<tr><td>SOAP_O:</td> <td><input type='text' name='SOAP_O'/></td></tr>");
echo("<tr><td>SOAP_A:</td> <td><input type='text' name='SOAP_A'/></td></tr>");
echo("<tr><td>SOAP_P:</td> <td><input type='text' name='SOAP_P'/></td></tr>");
// if there are nurses available
if ($nrows3 != 0)
{
  foreach($result3->fetchAll(PDO::FETCH_ASSOC) as $row3)
  {
    echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='nurse_list[]' value='{$row3['VAT']}'><label>{$row3['name']} {$row3['VAT']}</label></td></tr>");
  }
}
else
{
  echo("<tr><td>ATTENTION: </td><td>No nurse/assistant available for this consultation</td></tr>");
}
foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
{
  // check list of diagnostic codes
  $id = $diagnostic['ID'];
  echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='codes[]' value='{$diagnostic['ID']}'><label>{$diagnostic['ID']}</label></td>");
  // check list of medication per diagnostic
  if ($nrows_medication != 0)
  {
    echo("<td>Associated Prescriptions: </td></tr>");
    foreach($med_array as $med)
    {
      $med_name = $med['name'];
      $med_lab = $med['lab'];
      $value = serialize($med);
      echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name='meds_id[$id][]' value='$value'><label>{$med_name} {$med_lab}</label></td>");
      echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[$id][$med_name][$med_lab]'/></td>");
      echo("<td>Description:</td> <td><input type='text' name='description[$id][$med_name][$med_lab]'/></td></tr>");
    }
  }
  else
```

```
    {
      echo("<td>No Medication on the DataBase</td></tr>");
    }
  }
  echo("<tr><td><input type='submit' value ='Submit Information'/></td></tr>");
  echo("</form></table>");
}
// Consultation already exists
else
{
  foreach($result->fetchAll(PDO::FETCH_ASSOC) as $row)
  {
    echo("<form action='updateConsultation.php' method='post'>");
      echo("<input type='hidden' name='doctor' value='{$doctor}'>");
      echo("<input type='hidden' name='date' value='{$date}'>");
      echo("<tr><td>SOAP_S:</td> <td><input type='text' value='{$row['SOAP_S']}'
         name='SOAP_S'/></td></tr>");
      echo("<tr><td>SOAP_O:</td> <td><input type='text' value='{$row['SOAP_O']}'
         name='SOAP_O'/></td></tr>");
      echo("<tr><td>SOAP_A:</td> <td><input type='text' value='{$row['SOAP_A']}'
         name='SOAP_A'/></td></tr>");
      echo("<tr><td>SOAP_P:</td> <td><input type='text' value='{$row['SOAP_P']}'
         name='SOAP_P'/></td></tr>");
      //Nurses already assigned to this consultation
      if ($nrows2 != 0)
      {
        foreach($result2->fetchAll(PDO::FETCH_ASSOC) as $row2)
        {
          echo("<input type='hidden' name='nurses_preassigned[]' value='{$row2['VAT
             ']}'>");
          echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='
             nurses_assigned[]' value='{$row2['VAT']}' checked><label>{$row2['name
             ']} {$row2['VAT']}</label></td></tr>");
        }
      }
      if ($nrows3 != 0)
      {
        foreach($result3->fetchAll(PDO::FETCH_ASSOC) as $row3)
        {
          echo("<tr><td>Assistant Nurses: </td> <td><input type='checkbox' name='
             nurse_list[]' value='{$row3['VAT']}'><label>{$row3['name']} {$row3['
             VAT']}</label></td></tr>");
        }
      }
      elseif ($nrows2 == 0 and $nrows3 == 0)
      {
        echo("<tr><td>ATTENTION: </td><td>No nurse/assistant available for this
           consultation</td></tr>");
      }
      // already created diagnosis
      if($nrows_consult_diagnostic != 0)
      {
        foreach($result_consult_diagnostic->fetchAll(PDO::FETCH_ASSOC) as
           $diagnostic)
        {
          // check list of diagnostic codes
          $id = $diagnostic['ID'];
          echo("<input type='hidden' name='preassigned_codes[]' value='{$diagnostic
             ['ID']}'>");
          echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='
             assigned_codes[]' value='{$diagnostic['ID']}' checked><label>{
             $diagnostic['ID']}</label></td>");
          echo("<td>Associated Prescriptions: </td></tr>");
          // find medication already prescribed
          $result_medication2 = $connection->prepare("SELECT * FROM prescription
             WHERE VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
          $result_medication2->execute(array($doctor,$date,$id));
          $error6 = $result_medication2->errorInfo();
          if ($error6[1] != ''){
            echo("<p>Error: {$error6[2]}</p>");
          }

          $nrows_medication2 = $result_medication2->rowCount();
          if($nrows_medication2 != 0)
          {
            foreach($result_medication2->fetchAll(PDO::FETCH_ASSOC) as $med)
            {
              $med_name = $med['name'];
              $med_lab = $med['lab'];
              $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
              $value = serialize($prevalue);
              echo("<input type='hidden' name='meds_id_preassigned[$id][]' value='
                 $value'>");
              echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name
                 ='meds_id_assigned[$id][]' value='$value' checked><label>{
                 $med_name} {$med_lab}</label></td>");
              echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[
                 $id][$med_name][$med_lab]' value='{$med['dosage']}'/></td>");
              echo("<td>Description:</td> <td><input type='text' name='description[
                 $id][$med_name][$med_lab]' value='{$med['description']}'/></td></
                 tr>");
            }
          }
          // find medication not prescribed
```

```
          $result_medication2 = $connection->prepare("SELECT * FROM medication WHERE (name,lab) NOT IN (SELECT name, lab FROM prescription WHERE VAT_doctor = ? AND date_timestamp = ? AND ID = ?)");
          $result_medication2->execute(array($doctor,$date,$id));
          $error7 = $result_medication2->errorInfo();
          if ($error7[1] != ''){
            echo("<p>Error: {$error7[2]}</p>");
          }

          $nrows_medication2 = $result_medication2->rowCount();
          if($nrows_medication2 != 0)
          {
            foreach($result_medication2->fetchAll(PDO::FETCH_ASSOC) as $med)
            {
              $med_name = $med['name'];
              $med_lab = $med['lab'];
              $prevalue = array("name" => $med['name'], "lab" => $med['lab']);
              $value = serialize($prevalue);
              echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name='meds_id[$id][]' value='$value'><label>{$med_name} {$med_lab}</label></td>");
              echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[$id][$med_name][$med_lab]'/></td>");
              echo("<td>Description:</td> <td><input type='text' name='description[$id][$med_name][$med_lab]'/></td></tr>");
            }
          }
        }
      }
      // not yet created diagnosis
      foreach($result_diagcodes->fetchAll(PDO::FETCH_ASSOC) as $diagnostic)
      {
        // check list of diagnostic codes
        $id = $diagnostic['ID'];
        echo("<tr><td>Diagnostic Code: </td> <td><input type='checkbox' name='codes[]' value='{$diagnostic['ID']}'><label>{$diagnostic['ID']}</label></td>");
        // check list of medication per diagnostic
        if ($nrows_medication != 0)
        {
          echo("<td>Associated Prescriptions: </td></tr>");
          foreach($med_array as $med)
          {
            $med_name = $med['name'];
            $med_lab = $med['lab'];
            $value = serialize($med);
            echo("<tr><td></td><td></td><td></td> <td><input type='checkbox' name='
                meds_id[$id][]' value='$value'><label>{$med_name} {$med_lab}</label></td>");
            echo("<td></td><td>Dosage:</td> <td><input type='text' name='dosage[$id][$med_name][$med_lab]'/></td>");
            echo("<td>Description:</td> <td><input type='text' name='description[$id][$med_name][$med_lab]'/></td></tr>");
          }
        }
      }
      echo("<tr><td><input type='submit' value ='Submit Information'/></td></tr>");
    echo("</form></table>");
  }
}
```

3.4.1 Creating a Consultation

If the Consultation was not yet created the page created by the following code (Listing 8) will be used, this page will inform us how many rows were created.

Listing 8: Creating a Consultation (addConsultation.php)

```
$date = $_REQUEST['date'];
$doctor = $_REQUEST['doctor'];
$s = $_REQUEST['SOAP_S'];
$o = $_REQUEST['SOAP_O'];
$a = $_REQUEST['SOAP_A'];
$p = $_REQUEST['SOAP_P'];


$result = $connection->prepare("INSERT INTO consultation VALUES (?, ?, ?, ?, ?, ?)");
$result->execute(array($doctor,$date,$s,$o,$a,$p));

$error = $result->errorInfo();
if ($error[1] != ''){
  echo("<p>Error: {$error[2]}</p>");
}
else
{
  echo("<p>Consultation Created</p>");
}


if(!empty($_REQUEST['nurse_list']))
{
  foreach($_REQUEST['nurse_list'] as $nurse)
  {
```

```
  $result = $connection->prepare("INSERT INTO consultation_assistant VALUES (?, ?, ?)");
    $result->execute(array($doctor,$date,$nurse));

    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
  }
  else
  {
   echo("<p>Nurses/Assistants Assigned</p>");
  }
 }
}

if(!empty($_REQUEST['codes']))
{
 foreach($_REQUEST['codes'] as $code)
 {

  $result = $connection->prepare("INSERT INTO consultation_diagnostic VALUES (?, ?, ?)");
    $result->execute(array($doctor,$date,$code));

    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
  }
  else
  {
   echo("<p>Diagnosis Created</p>");
  }

  //insert medication for each diagnosis
  $med_id = $_REQUEST['meds_id'];
  if(!empty($med_id[$code]))
  {
   foreach($med_id[$code] as $med)
   {
    $med = unserialize($med);
    $name = $med['name'];
    $lab = $med['lab'];
    $dosage = $_REQUEST['dosage'];
    $dosage = $dosage[$code][$name][$lab];
    $description = $_REQUEST['description'];
    $description = $description[$code][$name][$lab];

```

```
        $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?,?, ?,?,?)");
        $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description));

        $error = $result->errorInfo();
        if ($error[1] != ''){
          echo("<p>Error: {$error[2]}</p>");
        }
        else
        {
          echo("<p>Prescription Created</p>");
        }

      }
    }
  }
}
```

3.4.2 Updating a Consultation

When updating a consultation, there's more details then when simply creating it, here we have to consider what was already assigned to this consultation, thus what will be deleted, inserted or simply updated.

The check lists on Listing 7 always send to this page three arrays for each one of the attributes (nurses, prescriptions, medication), one with what was previously on the database, a second with what the user wanted to keep on the database, and even another with new things that the user wants to insert. Taking the difference of the first two arrays, it's possible to figure out what should be deleted, and what should be untouched (except for the case of medication where we have to do an update). The other array contains the items to be inserted.

For the medication is a little bit different since it is possible for the dosage and/or description to have been updated. For this case, instead of ignoring the prescriptions that the user wanted to keep, we do an update with the information corresponding to all of those prescriptions.

Listing 9: Updating a Consultation (updateConsultation.php)

```
$date = $_REQUEST['date'];
$doctor = $_REQUEST['doctor'];
$s = $_REQUEST['SOAP_S'];
$o = $_REQUEST['SOAP_O'];
$a = $_REQUEST['SOAP_A'];
$p = $_REQUEST['SOAP_P'];

$result = $connection->prepare("UPDATE consultation SET SOAP_S = ?, SOAP_O = ?, SOAP_A = ?, SOAP_P = ? WHERE VAT_doctor = ? and date_timestamp = ?");
$result->execute(array($s,$o,$a,$p,$doctor,$date));

```

```
$error = $result->errorInfo();
if ($error[1] != ''){
  echo("<p>Error: {$error[2]}</p>");
}
else{
  echo("<p>Consultation updated</p>");
}

//nurses
if(!empty($_REQUEST['nurses_preassigned']))
{
  //removed nurses from consultation
  if(!empty($_REQUEST['nurses_assigned']))
  {
    $deletions = array_diff($_REQUEST['nurses_preassigned'],$_REQUEST['
       nurses_assigned']);
  }
  else
  {
    $deletions = $_REQUEST['nurses_preassigned'];
  }
  foreach($deletions as $deletion)
  {
    $result = $connection->prepare("DELETE FROM consultation_assistant WHERE
       VAT_doctor = ? and date_timestamp = ? and VAT_nurse = ?");
    $result->execute(array($doctor,$date,$deletion));
    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
    }
    else{
      echo("<p>Assistants Removed</p>");
    }
  }
}
if(!empty($_REQUEST['nurse_list']))
{
  //nurses added to consultation
  foreach($_REQUEST['nurse_list'] as $nurse)
  {
    $result = $connection->prepare("INSERT INTO consultation_assistant VALUES (?,
       ?, ?)");
    $result->execute(array($doctor,$date,$nurse));
    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
    }
```

```php
    else{
      echo("<p>New Nurses/Assistants Assigned</p>");
    }
  }
}

//new diagnosis
if(!empty($_REQUEST['codes']))
{
  foreach($_REQUEST['codes'] as $code)
  {
    $result = $connection->prepare("INSERT INTO consultation_diagnostic VALUES (?, ?, ?)");
    $result->execute(array($doctor,$date,$code));
    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
    }
    else{
      echo("<p>Diagnosis Created</p>");
    }

    //insert medication for each diagnosis
    $med_id = $_REQUEST['meds_id'];
    if(!empty($med_id[$code]))
    {
      foreach($med_id[$code] as $med)
      {
        $med = unserialize($med);
        $name = $med['name'];
        $lab = $med['lab'];
        $dosage = $_REQUEST['dosage'];
        $dosage = $dosage[$code][$name][$lab];
        $description = $_REQUEST['description'];
        $description = $description[$code][$name][$lab];

        $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?, ?, ?, ?, ?)");
        $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description));
        $error = $result->errorInfo();
        if ($error[1] != ''){
          echo("<p>Error: {$error[2]}</p>");
        }
        else{
          echo("<p>Prescriptions added</p>");
        }
      }
    }
  }
}

//already assigned diagnosis
if(!empty($_REQUEST['preassigned_codes']))
{
  //check changes on the already assigned diagnosis
  if(!empty($_REQUEST['assigned_codes'])){
    $deletions = array_diff($_REQUEST['preassigned_codes'],$_REQUEST['
       assigned_codes']);}
  else{
    $deletions = $_REQUEST['preassigned_codes'];}
  //deletions
  foreach($deletions as $code)
  {
    //delete all prescriptions for this consultation_diagnostic
    $result = $connection->prepare("DELETE FROM prescription WHERE VAT_doctor = ?
       AND date_timestamp = ? AND ID = ?");
    $result->execute(array($doctor,$date,$code));
    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
    }
    else{
      echo("<p>Prescriptions Deleted</p>");
    }

    //delete consultation_diagnostic
    $result = $connection->prepare("DELETE FROM consultation_diagnostic WHERE
       VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
    $result->execute(array($doctor,$date,$code));
    $error = $result->errorInfo();
    if ($error[1] != ''){
      echo("<p>Error: {$error[2]}</p>");
    }
    else{
      echo("<p>Diagnosis Deleted</p>");
    }
  }
  //updates on preassigned diagnosis
  $meds_assigned = $_REQUEST['meds_id_assigned'];
  $meds_preassigned = $_REQUEST['meds_id_preassigned'];
  foreach($_REQUEST['assigned_codes'] as $code)
  {
    if(!empty($meds_preassigned[$code]))
    {
      if(!empty($meds_assigned[$code])){
        //updates to be made
        foreach($meds_assigned[$code] as $update)
        {
          $med = unserialize($update);
          $name = $med['name'];
          $lab = $med['lab'];
          $dosage = $_REQUEST['dosage'];
          $dosage = $dosage[$code][$name][$lab];
          $description = $_REQUEST['description'];
          $description = $description[$code][$name][$lab];

          $result = $connection->prepare("UPDATE prescription SET dosage = ?,
             description = ? WHERE name = ? AND lab = ? AND VAT_doctor = ? AND
             date_timestamp = ? AND ID = ?");
          $result->execute(array($dosage,$description,$name,$lab,$doctor,$date,
             $code));
          $error = $result->errorInfo();
          if ($error[1] != ''){
            echo("<p>Error: {$error[2]}</p>");
          }
          else{
            echo("<p>Updates Completed</p>");
          }
        }
          $deletions = array_diff($meds_preassigned[$code],$meds_assigned[$code]);}
      else{
        $deletions = $meds_preassigned[$code];}
      //deletions to be made
      foreach($deletions as $deletion)
      {
        $med = unserialize($deletion);
        $name = $med['name'];
        $lab = $med['lab'];


        $result = $connection->prepare("DELETE FROM prescription WHERE name = ? AND
            lab = ? AND VAT_doctor = ? AND date_timestamp = ? AND ID = ?");
        $result->execute(array($name,$lab,$doctor,$date,$code));
        $error = $result->errorInfo();
        if ($error[1] != ''){
          echo("<p>Error: {$error[2]}</p>");
        }
        else{
          echo("<p>Prescriptions Deleted</p>");
        }
      }
    }
    //insertions of meds for this diagnostic
    $med_id = $_REQUEST['meds_id'];
    if(!empty($med_id[$code]))
    {
      foreach($med_id[$code] as $med)
      {
        $med = unserialize($med);
        $name = $med['name'];
        $lab = $med['lab'];
        $dosage = $_REQUEST['dosage'];
        $dosage = $dosage[$code][$name][$lab];
        $description = $_REQUEST['description'];
        $description = $description[$code][$name][$lab];

        $result = $connection->prepare("INSERT INTO prescription VALUES (?, ?, ?,
           ?, ?,?,?)");
        $result->execute(array($name,$lab,$doctor,$date,$code,$dosage,$description)
           );
        $error = $result->errorInfo();
        if ($error[1] != ''){
          echo("<p>Error: {$error[2]}</p>");
        }
        else{
          echo("<p>Prescriptions added</p>");
        }
      }
    }
  }
}
```

4 Web Application Using the Database - 3/3

4.1 Add procedure in consultation

On the previous section, it was created a page where the information of a consultation would be displayed. To this page, it was added a button where the user can add a particular type of procedure (in this case a dental chart), this button will redirect the user to a new page where the query will be done (*dental_ charting.php*).

4.2 Charting measures

On this page, *dental_ charting.php*, the user can insert the results of the dental charting, i.e. per-teeth measurements for the gap between the tooth and the gums and a description. For each dental chart, all the measures will be made in the context of the same transaction (listing 4.2)

```php
$sql = "SELECT * from teeth";
$result = $connection->query($sql);
$nrows = $result->rowCount();
if ($nrows == 0)
{
echo("<p>erro!</p>");
}
else
{
    echo("<form action='dental_charting.php' method='post'> <table>");
    echo("<tr> <th>Name</th> <th>Quadrant</th> <th>Number</th> <th>Description</th> <th>Measure</th> <th></th> </tr>");
    foreach($result as $row)
    {
        //insert description array
        echo("<tr> ");
        echo("<td>{$row['name']}</td> <td>{$row['quadrant']} </td> <td>{$row['number']} </td>");
        echo("<td>  <input type='text' name='description[{$row['quadrant']}][{$row['number']}]'/>  </td>");
        echo("<td>  <input type='number' step ='0.1' name='measure[{$row['quadrant']}][{$row['number']}]' min = '0'/>  </td>");
        echo("<input type='hidden' value='$doctor' name='doctor' >");
        echo("<input type='hidden' value='$name' name='name' >");
        echo("<input type='hidden' value='$date' name='date' >");
        echo(" </tr>");

    }
    echo("</table> <p><input type='submit' value='new measure'> </form>");
}

$description = $_REQUEST['description'];
$measure = $_REQUEST['measure'];
$desc = 'default';


$result = $connection->prepare("insert into procedure_in_consultation values (?,?,?,?)");
$result->execute(array($name,$doctor,$date,$desc));
$error = $result->errorInfo();
if ($error[1] == ''){
    echo("<p>procedure in consultation added</p>");
}

try{
    $connection->beginTransaction();

    for($i=1;$i<=2;$i++){
        for($j=1;$j<=3;$j++){
            if(!empty($measure[$i][$j])){

                $result = $connection->prepare("insert into procedure_charting values (?, ?, ?,?,?,?,?)");
                $result2 = $connection->prepare("UPDATE procedure_charting SET description = ?, measure = ? WHERE name = ? and VAT = ? and date_timestamp = ? and quadrant=? and number = ?");

                if($result->execute(array($name, $doctor, $date,$i,$j,$description[$i][$j],$measure[$i][$j]))){
                    echo("<p>New measure added</p>");
                }
                else if ($result2->execute(array($description[$i][$j],$measure[$i][$j],$name, $doctor, $date,$i,$j))){
                    echo("<p>New measure updated</p>");

                }
                else{
                    $error = $result->errorInfo();
                    if ($error[1] != ''){
                        echo("<p>Error: {$error[2]}</p>");
                    }
                }
            }
        }
    }

    $connection->commit();
} catch(Exception $e) {
    $connection->rollback();
    throw $e;
  }


$connection = null;
```

First of all, a query is done to print on the page all the tooth and make it possible to add information assigned to each teeth. The information regarding each teeth is stored in a 2D array according to its quadrant and number.

After that, the procedure in consultation is inserted in the database.

At last, a transaction is started and all the measures inserted will be committed at the same time. If by any chance, already exists a measure for that teeth, instead of a insert into the database it will be done an update

5 Functions, Triggers and Stored Procedures

5.1 Trigger: Age Update

For the age update, the *timestampdiff()* function does the trick, subtracting the birth date from the current date and returning the value in years. So every time there is an insert on appointment for a certain client, this calculation is made, updating the age of that specific client.

Listing 10: Age Update

```sql
delimiter $$
create trigger update_age after insert on appointment
for each row
begin
  update client
  set age = timestampdiff(year, birth_date, now())
  where VAT = new.VAT_client;
end $$
delimiter ;
```

5.2 Trigger: Same individual, Multiple jobs

5.2.1 (a) Nurses, Receptionists and Doctors

In other to prevent nurses and receptionists of also being doctors, and vice-versa, there were created triggers that fire whenever there are an update or an insert on one of this tables, that's why there are two triggers for each table. The triggers simply signal an error that aborts the query whenever one of this prohibited actions occur.

Listing 11: (a) Nurses and Receptionists can't be Doctors

```sql
-- insert nurse
delimiter $$
create trigger check_nurse_insert before insert on nurse
for each row
begin
  if exists(select VAT
  from doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the same VAT';
  end if;
end $$
delimiter ;
-- update nurse
delimiter $$
create trigger check_nurse_update before update on nurse
for each row
begin
```

```
  if exists(select VAT
  from doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the same VAT';
  end if;
end $$
delimiter ;
-- insert receptionist
delimiter $$
create trigger check_receptionist_insert before insert on receptionist
for each row
begin
  if exists(select VAT
  from doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the same VAT';
  end if;
end $$
delimiter ;
-- update receptionist
delimiter $$
create trigger check_receptionist_update before update on receptionist
for each row
begin
  if exists(select VAT
  from doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a doctor with the same VAT';
  end if;
end $$
delimiter ;
-- insert doctor
delimiter $$
create trigger check_doctor_insert before insert on doctor
for each row
begin
  if exists(select VAT
  from nurse where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a nurse with the same VAT';
  end if;
  if exists(select VAT
  from receptionist where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a receptionist with the same VAT';
  end if;
end $$
```

```
delimiter ;
-- update doctor
delimiter $$
create trigger check_doctor_update before update on doctor
for each row
begin
  if exists(select VAT
  from nurse where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a nurse with the same VAT';
  end if;
  if exists(select VAT
  from receptionist where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a receptionist with the same VAT';
  end if;
end $$
delimiter ;
```

5.2.2 (b) Permanent Doctors vs Trainees

What was explained on (a) happens here, except for the fact that the considered tables are the one for permanent doctors, and the one for trainees.

Listing 12: (b) Doctors can't be both Permanent and Trainees

```
-- insert permanent_doctor
delimiter $$
create trigger check_permanent_insert before insert on permanent_doctor
for each row
begin
  if exists(select VAT
  from trainee_doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a trainee with the same VAT';
  end if;
end $$
delimiter ;
-- update permanent_doctor
delimiter $$
create trigger check_permanent_update before update on permanent_doctor
for each row
begin
  if exists(select VAT
  from trainee_doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a trainee with the same VAT';
  end if;
end $$
delimiter ;
-- insert trainee_doctor
delimiter $$
create trigger check_trainee_insert before insert on trainee_doctor
for each row
begin
  if exists(select VAT
  from permanent_doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a permanent doctor with the same VAT';
  end if;
end $$
delimiter ;
-- update trainee_doctor
delimiter $$
create trigger check_trainee_update before update on trainee_doctor
for each row
begin
  if exists(select VAT
  from permanent_doctor where VAT = new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'There is already a permanent doctor with the same VAT';
  end if;
end $$
delimiter ;
```

5.3 Trigger: Repeated Phone Numbers

Since we can't allow repeated phone numbers, it is necessary to fire an error signal whenever there are an insertion or an update on the table for employees' phone numbers and on the table for clients' phone numbers. That being said, the rationale behind the trigger is the same as in the previous section.

Listing 13: Triggers for Repeated Phone Numbers

```
-- on insert phone
delimiter $$
create trigger check_phone_insert before insert on phone_number_employee
for each row
begin
  if exists(select phone
  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a employee';
  end if;
```

```sql
  if exists(select phone
  from phone_number_client where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a client';
  end if;
end $$
delimiter ;
-- on update phone
delimiter $$
create trigger check_phone_update before update on phone_number_employee
for each row
begin
  if exists(select phone
  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a employee';
  end if;
  if exists(select phone
  from phone_number_client where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a client';
  end if;
end $$
delimiter ;
-- on insert phone
delimiter $$
create trigger check_phone_insert_client before insert on phone_number_client
for each row
begin
  if exists(select phone
  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a employee';
  end if;
  if exists(select phone
  from phone_number_client where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a client';
  end if;
end $$
delimiter ;
-- on update phone
delimiter $$
create trigger check_phone_update_client before update on phone_number_client
for each row
begin
  if exists(select phone
  from phone_number_employee where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a employee';
  end if;
  if exists(select phone
  from phone_number_client where phone = new.phone and VAT != new.VAT) then
  signal sqlstate '45000' set MESSAGE_TEXT = 'This phone number already exists, belongs to a client';
  end if;
end $$
delimiter ;
```

5.4 Function: Number of *No Shows*

This function aims to compute the number of appointments that did not result in a consultation given some parameters, in order to do that it's necessary to first search every appointment that fulfils those requisitions. After that just count the number of appointments whose primary keys do not exist on the consultations table.

Listing 14: No Shows

```
delimiter $$
create function no_shows(c_gender char(2), a_year varchar(4), max_age int, min_age int)
returns integer
begin
  declare ns_count integer;
  select count(*) into ns_count
  from appointment
  where VAT_client in (select VAT
  from client where
  age between max_age and min_age
  and gender = c_gender)
  and (VAT_doctor, date_timestamp) not in
  (select VAT_doctor, date_timestamp from consultation)
  and extract(year from date_timestamp) = a_year;
  return ns_count;
end$$
delimiter ;
```

5.5 Procedure: Salary Change

On this procedure we do two updates, one for doctors that had more than 100 consultations, and the other for the opposite case, besides that we just have to check the number of years of practice for each permanent doctor and consider only the consultations for the current year.

Listing 15: Doctors Salary Change

```
delimiter $$
create procedure salary_raise(in x_years integer)
begin
  update employee
  set salary = salary + salary*0.05
  where VAT in (select VAT
  from permanent_doctor
  where years > x_years)
  and VAT not in (select VAT_doctor
  from consultation
  where extract(year from date_timestamp) = year(curdate())
  group by VAT_doctor
  having count(*) > 100);

  update employee
  set salary = salary + salary*0.1
  where VAT in (select VAT
  from permanent_doctor
  where years > x_years)
  and VAT in (select VAT_doctor
  from consultation
  where extract(year from date_timestamp) = year(curdate())
  group by VAT_doctor
  having count(*) > 100);
end$$
delimiter ;
```