Instituto Superior Técnico

Information Systems and Databases

Project Assignment - Part 3

Assignment 3 - Using the Database

Alunos	Número de aluno
Bernardo Bastos	84012
Ivo Simões	87024
João Chumbinho	81761
Group	32

1 A Web Application Using the Database

1.1 1st Task

In all pages it is needed a connection to the database. For that, in all of them is included a php file that is used to make the connection. The source code is shown below.

```php
<?php
    $host = "db.tecnico.ulisboa.pt";
    $user = "ist187024";
    $pass = "pass";
    $dsn = "mysql:host=$host;dbname=$user";
    try
    {
        $connection = new PDO($dsn, $user, $pass);
    }
    catch(PDOException $exception)
    {
        echo("<p>Error: ");
        echo($exception->getMessage());
        echo("</p>");
    exit();
    }
?>
```

The first page is to search the clients in the database. The results are shown in other page depending on what the user searched for. In the search, if the VAT is provided, it will only use the VAT to find a client that match that value. If it is not provided, the search will take the name and address (Zip, City, Street) (or part of them), and perform the search. The prints of the page and its respective code are represented below.

(a) Search page with no inputs

(b) Search page with VAT input

(c) Search page with a partial name, zip, city and street

Figure 1: Search page

The source code for the search page is represented.

```html
<html>
    <body>

        <form action="display_clients.php" method="post">
        <h3>Search client</h3>
        <p>VAT: <input type="number" name="VAT" pattern="\d"/></p>
        <p>Name: <input type="text" name="name_"/></p>
        <p>Street: <input type="text" name="street"/></p>
        <p>City: <input type="text" name="city"/></p>
        <p>Zip: <input type="text" name="zip"/></p>

<?php
    include 'connect_db.php';
?>
        <p><input type="submit" value="Submit"/></p>
        </form>
    </body>
</html>
```
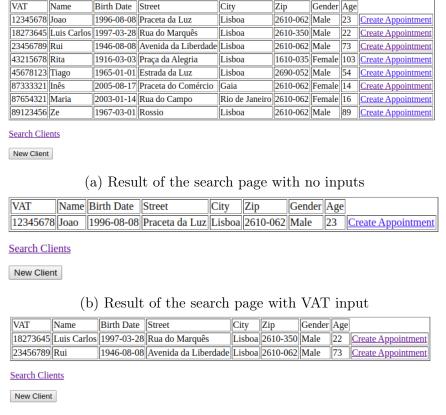
The results of the previous searches are represented in the page to display the clients in figure 2.

VAT	Name	Birth Date	Street	City	Zip	Gender	Age	
12345678	Joao	1996-08-08	Praceta da Luz	Lisboa	2610-062	Male	23	Create Appointment
18273645	Luis Carlos	1997-03-28	Rua do Marquês	Lisboa	2610-350	Male	22	Create Appointment
23456789	Rui	1946-08-08	Avenida da Liberdade	Lisboa	2610-062	Male	73	Create Appointment
43215678	Rita	1916-03-03	Praça da Alegria	Lisboa	1610-035	Female	103	Create Appointment
45678123	Tiago	1965-01-01	Estrada da Luz	Lisboa	2690-052	Male	54	Create Appointment
87333321	Inês	2005-08-17	Praceta do Comércio	Gaia	2610-062	Female	14	Create Appointment
87654321	Maria	2003-01-14	Rua do Campo	Rio de Janeiro	2610-062	Female	16	Create Appointment
89123456	Ze	1967-03-01	Rossio	Lisboa	2610-062	Male	89	Create Appointment

Search Clients

New Client

(a) Result of the search page with no inputs

VAT	Name	Birth Date	Street	City	Zip	Gender	Age	
12345678	Joao	1996-08-08	Praceta da Luz	Lisboa	2610-062	Male	23	Create Appointment

Search Clients

New Client

(b) Result of the search page with VAT input

VAT	Name	Birth Date	Street	City	Zip	Gender	Age	
18273645	Luis Carlos	1997-03-28	Rua do Marquês	Lisboa	2610-350	Male	22	Create Appointment
23456789	Rui	1946-08-08	Avenida da Liberdade	Lisboa	2610-062	Male	73	Create Appointment

Search Clients

New Client

(c) Result of the search page with a partial name, zip, city and street

Figure 2: Search page

In this page, it receives the search parameters from the search page and performs a query dependent on the VAT input. If the VAT is empty, it performs a query that only takes the parameters from the name and address, otherwise it only uses the VAT. The source code is below.

```php
<html>
    <body>

<?php
    include 'connect_db.php';

    $VAT = $_REQUEST['VAT'];
    $name_ = $_REQUEST['name_'];
    $street = $_REQUEST['street'];
    $city = $_REQUEST['city'];
```

```php
11    $zip = $_REQUEST['zip'];

12

13    if(empty($VAT))
14    {
15        $sql = "SELECT * FROM client WHERE name_ LIKE ?
16        AND street LIKE ? AND city LIKE ? AND zip LIKE ?";
17        $stat = $connection->prepare($sql);

18

19        //Checks if error exists in prepare
20        if (false===$stat)
21        {
22            echo('prepare() failed: ');
23            exit();
24        }

25

26        $sth = $stat->execute(['%' . $name_ . '%', '%' . $street . '%',
    '%' . $city . '%', '%' . $zip . '%']);
27    }
28    else
29    {
30        $sql = "SELECT * FROM client WHERE VAT = ?";
31        $stat = $connection->prepare($sql);

32

33        //Checks if error exists in prepare
34        if (false===$stat)
35        {
36            echo('prepare() failed: ');
37            exit();
38        }

39

40        $sth=$stat->execute([$VAT]);
41    }

42

43    //Checks if error exists in execute
44    if ($sth == FALSE)
45    {
46      $info = $stat->errorInfo();
47      echo("<p>Error exec: {$info[2]}</p>");
48      exit();
49    }

50

51    echo("<table border=\"1\">");
```

```php
52      echo("<tr><td>VAT</td><td>Name</td><td>Birth Date</td><td>Street</td
    ><td>City</td><td>Zip</td><td>Gender</td><td>Age</td></tr>");
53
54      $result = $stat->fetchAll();
55      foreach($result as $row)
56      {
57          echo("<tr><td>");
58          echo($row['VAT']);
59          echo("</td><td>");
60          echo($row['name_']);
61          echo("</td><td>");
62          echo($row['birth_date']);
63          echo("</td><td>");
64          echo($row['street']);
65          echo("</td><td>");
66          echo($row['city']);
67          echo("</td><td>");
68          echo($row['zip']);
69          echo("</td><td>");
70          echo($row['gender']);
71          echo("</td><td>");
72          echo($row['age']);
73          echo("</td><td>");
74          echo("<a href=\"select_date_appointment.php?VAT_client=");
75          echo($row['VAT']);
76          echo("\">Create Appointment</a>\n");
77          echo("</td><td>");
78          echo("<a href=\"appointment_info.php?VAT_client=");
79          echo($row['VAT']);
80          echo("\">Info Appointment</a>\n");
81          echo("</td></tr>");
82      }
83      echo("</table>");
84
85      echo("<p><a href=\"search_clients.php\">Search Clients</a></p>");
86      $connection = null;
87  ?>
88
89          <form action="insert_new_client.php" method="post">
90          <p><input type="submit" value="New Client"/></p>
91          </form>
92
```

```
93        </body >
94  </html >
```

In the page where the clients are displayed there is an option to create a new client. All the fields are required, which was implemented with HTML input. The print of the page can be observed in figure 3.

New Client

VAT: 12298876

Name: Tomas Oliveira

Birth Date: 07/09/1998

Street: Rua Jose Alberto

City: Porto

Zip: 4000-011

Gender: Male ▼

Age: 21

Phone Number: 351912323215

Submit

Figure 3: Insert new client page

The source code for this page is the following:

```
1   <html >
2       <body >
3
4           <form action="client_created.php" method="post">
5           <h3>New Client</h3>
6           <p>VAT: <input type="text" name="VAT" required/></p>
7           <p>Name: <input type="text" name="name_" required/></p>
8           <p>Birth Date: <input type="date" name="birth_date" min="
       1900-01-01" max="<?php echo date('Y-m-d');?>" required/></p>
9           <p>Street: <input type="text" name="street" required/></p>
10          <p>City: <input type="text" name="city" required/></p>
11          <p>Zip: <input type="text" name="zip" required/></p>
```

```
12          <p>Gender: <select  name="gender" required>
13                    <option value="male">Male</option>
14                    <option value="female">Female</option>
15              </select></p>
16
17          <p>Age: <input type="number" name="age" max="119" required/></p>
18          <p>Phone Number: <input type="number" name="phone_number"
    required/></p>
19
20  <?php
21      include 'connect_db.php';
22  ?>
23          <p><input type="submit" value="Submit"/></p>
24          </form>
25      </body>
26  </html>
```

After inserting a new client, this page redirects to a page where it displays the newly inserted client.

VAT	Name	Birth Date	Street	City	Zip	Gender	Age	Phone Number	
12298876	Tomas Oliveira	1998-07-09	Rua Jose Alberto	Porto	4000-011	male	21	351912323215	Create Appointment

Search Clients

Figure 4: Client inserted page

Both, in the pages where the clients are displayed after the search, or when you insert a new client, is possible to create an appointment. That link redirects to a page to select the date for the appointment. In this page to create appointments in future dates, implemented with HTML input, and it is only possible to choose hours from 9h to 17h. To implement this feature it was used a HTML select together with PHP code, where it iterates an integer between 9 and 17 and concatenates a string to format to a hour. The print of the page is below.

Choose a date for the appointment

Date: 12/20/2019

Time: 15:00:00 ▾

Check Available Doctors

Figure 5: Select date for appointment page

The source code for this page is below.

```
1  <html>
2      <body>
3          <form action="select_doctor.php?" method="post">
4          <h3>Choose a date for the appointment</h3>
5          <p>Date: <input type="date" name="date" min="<?php echo date('Y-
   m-d');?>" required/></p>
6          <p>Time:
7              <select name="time">
8  <?php
9      include 'connect_db.php';
10
11     $VAT_client = $_REQUEST['VAT_client'];
12
13     for($hours=9; $hours<18; $hours++)
14     {
15         $time = str_pad($hours,2,'0',STR_PAD_LEFT).':00:00';
16         echo '<option value= "'.$time.'">'.$time.'</option>';
17     }
18 ?>
19              </select>
20          </p>
21          <input type="hidden" name="VAT_client" value=<?=$_REQUEST['
   VAT_client']?>>
22          <p><input type="submit" value="Check Available Doctors"/></p>
23          </form>
24
25      </body>
26 </html>
```

The next page shows all the doctors that are available for the chosen date. This page has a query that selects all doctors that do not have appointments in the chosen

date.

Available Doctors

VAT	Name	
11111222	Jack Sweettath	Create Appointment
12345876	Joao	Create Appointment
12456876	Rui	Create Appointment
14444876	Richard	Create Appointment
44440000	Joseph	Create Appointment
55555555	Mara	Create Appointment
88888888	Jane Sweettooth	Create Appointment
222000333	Johnson	Create Appointment

Figure 6: Available doctors page

The source code of this page is below.

```
1  <html>
2      <body>
3  <?php
4      include 'connect_db.php';
5
6      $date = $_REQUEST['date'];
7      $time = $_REQUEST['time'];
8      $VAT_client = $_REQUEST['VAT_client'];
9
10     $date_timestamp = strval($date).' '. strval($time);
11
12
13     $sql = "SELECT e.name_, e.VAT FROM employee as e, doctor as d WHERE
       e.VAT NOT IN
14            (SELECT VAT_doctor FROM appointment as a
15            WHERE a.date_timestamp = '$date_timestamp') AND e.VAT = d.
       VAT";
16
17     $stat = $connection->prepare($sql);
18
19     //Checks if error exists in prepare
20     if (false===$stat)
21     {
22         echo('prepare() failed: ');
23         exit();
24     }
```

```
25
26     $sth=$stat ->execute ([$date_timestamp]);
27
28     //Checks if error exists in execute
29     if ($sth == FALSE)
30     {
31       $info = $stat ->errorInfo ();
32       echo("<p>Error exec: {$info[2]}</p>");
33       exit ();
34     }
35
36     echo("<h3>Available Doctors </3>");
37     echo("<table border =\"1\">");
38     echo("<tr><td>VAT </td><td>Name </td></tr>");
39
40     $result = $stat ->fetchAll ();
41     foreach ($result as $row)
42     {
43         echo("<tr><td>");
44         echo($row['VAT']);
45         echo("</td><td>");
46         echo($row['name_']);
47         echo("</td><td>");
48         echo("<a href =\"add_description.php?VAT_doctor =". $row['VAT'] ."
    &". "date_timestamp =". $date_timestamp ."&". "VAT_client =".
    $VAT_client);
49         echo("\">Create Appointment </a>\n");
50         echo("</td></tr>");
51     }
52     echo("</table >");
53     $connection = null;
54 ?>
55     </body >
56 </html >
```

The next page is used to add a description to the appointment, which can be observed in figure 7. In this page it receives the text input from the HTML and sends to the next page (together with other variables from the pages before like the VAT of doctor, the VAT of the client and the date and time of the appointment), to insert the appointment in the database.

Add a description for the appointment

Appointment with doctor Rui

Add description

Figure 7: Add description to the appointment page

The respective code is also present.

```html
<html>
    <body>
        <form action="create_appointment.php?" method="post">
        <h3>Add a description for the appointment</h3>
        <p><input type="test" name="description" maxlength="255"/></p>

<?php
    include 'connect_db.php';
?>
        </p>
        <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
        <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
    VAT_doctor']?>">
        <input type="hidden" name="date_timestamp" value="<?=$_REQUEST['
    date_timestamp']?>">
        <p><input type="submit" value="Add description"/></p>
        </form>

    </body>
</html>
```

In the end, it is shown to the user the appointment created, with possibility to return to the search clients page. This page, inserts into an appointment with all the variables coming from the previous pages.

Appointment Created

VAT Doctor	Date Time	Description	VAT Client
12456876	2019-12-20 15:00:00	Appointment with doctor Rui	12298876

Search Clients

Figure 8: Appointment created page

The code is for this page is below.

```php
1   <html>
2       <body>
3
4   <?php
5       include 'connect_db.php';
6
7       $VAT_doctor = $_REQUEST['VAT_doctor'];
8       $date_timestamp = $_REQUEST['date_timestamp'];
9       $VAT_client = $_REQUEST['VAT_client'];
10      $description = $_REQUEST['description'];
11
12      $sql = "INSERT INTO appointment VALUES(?,?,?,?)";
13      $stat = $connection->prepare($sql);
14
15      //Checks if error exists in prepare
16      if (false===$stat)
17      {
18          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
19      }
20
21      $result = $stat->execute([$VAT_doctor, $date_timestamp, $description
    , $VAT_client]);
22
23      //Checks if error exists in execute
24      if ($result == FALSE)
25      {
26        $info = $stat->errorInfo();
27        echo("<p>Error exec: {$info[2]}</p>");
28        exit();
29      }
30
31      echo("<h3>Appointment Created</h3>");
```

```
32    echo("<table border=\"1\">");
33    echo("<tr><td>VAT Doctor</td><td>Date Time</td><td>Description</td><
      td>VAT Client</td></tr>");
34
35    echo("<tr><td>");
36    echo($VAT_doctor);
37    echo("</td><td>");
38    echo($date_timestamp);
39    echo("</td><td>");
40    echo($description);
41    echo("</td><td>");
42    echo($VAT_client);
43    echo("</td></tr>");
44    echo("</table>");
45
46    echo("<p><a href=\"search_clients.php\">Search Clients</a></p>");
47 ?>
48    </body>
49 </html>
```

1.2   2nd Task

In pages were connection to the database is required the php file "connection.php" is included as in section 1.1.

In this task the following pages were developed do access and edit information related to an appointment/consultation for a given client obtained in the previous point (VAT_client is received from previous page). The information is divided into appointments and consultations (since these two type of events store different information), and presented chronologically.

The table related to the appointment list has two more columns which inform the user if the appointment is already or not a consultation. If not, the user is able to add more information to the appointment converting it into a consultation.

This page is represented in figure 9.

Consultations/Appointments for Client = 87333321

Appointments

Data	More Info	Appointment Type	Add More Info
2019-12-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-07-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-07 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-05 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-04 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-03 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-02 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-01 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-01-31 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-01-21 17:00:00	Get Info	Not a Consultation	+ADD Info
2010-12-06 17:00:00	Get Info	Not a Consultation	+ADD Info
2008-12-06 17:00:00	Get Info	Already a Consultation	Cant Edit

Consultations

Data	
2019-12-06 17:00:00	Get Info
2019-07-06 17:00:00	Get Info
2019-02-07 17:00:00	Get Info
2019-02-06 17:00:00	Get Info
2008-12-06 17:00:00	Get Info

(a) Result presenting Appointments and Consultations for Client 87333321

Appointments

Data	More Info	Appointment Type	Add More Info
2019-12-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-07-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-07 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-06 17:00:00	Get Info	Already a Consultation	Cant Edit
2019-02-05 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-04 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-03 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-02 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-02-01 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-01-31 17:00:00	Get Info	Not a Consultation	+ADD Info
2019-01-21 17:00:00	Get Info	Not a Consultation	+ADD Info
2010-12-06 17:00:00	Get Info	Not a Consultation	+ADD Info
2008-12-06 17:00:00	Get Info	Already a Consultation	Cant Edit

(b) Appointment Type and Add more Info

Figure 9: Search page

The source code for this page is presented in the following figure:

```html
<html>
  <head>
    <title>Appointment/Consultation Info</title>
  </head>
<body>
  <h2>Consultations/Appointments for Client = <?=$_REQUEST['VAT_client'
  ]?></h2>
  <h3>Appointments</h3>
    <?php

    include 'connect_db.php';

    $sql = "SELECT a.VAT_doctor,a.date_timestamp, c.SOAP_S FROM
```

```php
    appointment as a
13      LEFT JOIN consultation as c ON c.VAT_doctor = a.VAT_doctor AND c.
    date_timestamp = a.date_timestamp
14      WHERE a.VAT_client= ?
15      ORDER BY a.date_timestamp DESC";
16
17      $VAT_client_SELECTED = $_REQUEST['VAT_client'];
18
19      $stat = $connection->prepare($sql);
20      // prepare() can fail because of syntax errors, missing privileges
    , ....
21      if (false===$stat)
22      {
23          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
24      }
25
26      $result = $stat->execute([$VAT_client_SELECTED]);
27      if ($result == FALSE)
28      {
29          $info = $connection->errorInfo();
30          echo("<p>Error: {$info[2]}</p>");
31          exit();
32      }
33
34      echo("<table border=\"1\">");
35      echo("<tr><td>Data</td><td>More Info</td><td>Appointment Type</td
    ><td>Add More Info</td></tr>");
36      //Fetch the result
37      $result = $stat->fetchAll();
38      foreach($result as $row)
39      {
40          echo("<tr><td>");
41          echo($row['date_timestamp']);
42          echo("</td><td>");
43          echo("<a href=\"info_appointment.php?VAT_doctor=" . $row['
    VAT_doctor'] . "&date_timestamp=" . $row['date_timestamp'] ."&
    VAT_client=" . $VAT_client_SELECTED);
44          echo("\">Get Info</a>\n");
45
46          if(is_null($row['SOAP_S'])){
47              echo("</td><td>");
48              echo("Not a Consultation");
```

```php
49            echo("</td><td>");
50            echo("<a href=\"add_info_consultation_pre.php?VAT_doctor=" .
   $row['VAT_doctor'] . "&date_timestamp=" . $row['date_timestamp']  ."&
   VAT_client=" . $VAT_client_SELECTED);
51            echo("\">+ADD Info</a></td>\n");
52          }
53          else{
54            echo("</td><td>");
55            echo("Already a Consultation");
56            echo("</td><td>");
57            echo("Cant Edit</td>");
58          }
59          echo("</tr>");
60        }
61        echo("</table>");
62        $connection = null;
63      ?>
64 <h3>Consultations</h3>
65 <?php
66    include 'connect_db.php';
67
68    $sql = "SELECT * FROM appointment as a
69    INNER JOIN consultation as c ON a.VAT_doctor = c.VAT_doctor AND a.
     date_timestamp = c.date_timestamp
70    WHERE a.VAT_client= ?
71    ORDER BY c.date_timestamp DESC";
72
73    $stat = $connection->prepare($sql);
74    // prepare() can fail because of syntax errors, missing privileges,
     ....
75    if (false===$stat)
76    {
77       die('prepare() failed: ' . htmlspecialchars($mysqli->error));
78    }
79
80    $VAT_client_SELECTED = $_REQUEST['VAT_client'];
81
82    $result = $stat->execute([$VAT_client_SELECTED]);
83    if ($result == FALSE)
84    {
85      $info = $connection->errorInfo();
86      echo("<p>Error: {$info[2]}</p>");
```

```php
87      exit();
88    }
89    echo("<table border=\"1\">");
90    echo("<tr><td>Data</td></tr>");
91
92    //Fetch the result
93    $result = $stat->fetchAll();
94    foreach($result as $row)
95    {
96      echo("<tr><td>");
97      echo($row['date_timestamp']);
98      echo("</td><td>");
99      echo("<a href=\"info_consultation.php?VAT_doctor=" . $row['
        VAT_doctor'] . "&date_timestamp=" . $row['date_timestamp']  ."&
        VAT_client=" . $VAT_client_SELECTED);
100     echo("\">Get Info</a></td>\n");
101     echo("</tr>");
102   }
103   echo("</table>");
104
105   echo("<p><a href=\"search_clients.php\">Search Clients</a></p>");
106
107   $connection = null;
108 ?>
109  </body>
110 </html>
```

Since appointments and consultations store different information, different pages were created in order to present that information. Both pages have the action "Back to Appointment/Consultation List" to return to the previous page.

To see all the info related to a specific appointment the user must select the reference "Get Info" on the Appointments list. Once selected this page is redirected to the page "info_appointment.php" which receives the VAT_client and the date_time_stamp and presents the information presented in figure 10.
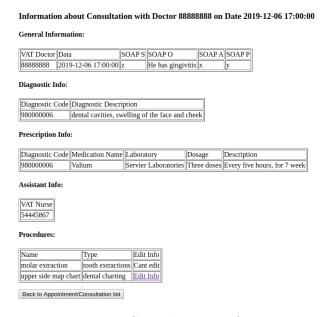
Information about Appointment with Doctor 88888888 on Date 2019-02-07 17:00:00

General Information:

VAT Doctor	Date	Description	VAT Client
88888888	2019-02-07 17:00:00	Não veio	87333321

Back to Appointment/Consultation list

Figure 10: Appointment Info page

The source code for this page is presented in the following figure:

```
1   <html>
2     <head>
3       <title>Appointment Info</title>
4     </head>
5      <body>
6      <h3>Information about Appointment with Doctor <?=$_REQUEST['
       VAT_doctor']?> on Date <?=$_REQUEST['date_timestamp']?></h3>
7      <h4>General Information:</h4>
8      <?php
9        include 'connect_db.php';
10
11       $sql = "SELECT * FROM appointment WHERE VAT_doctor= ? AND
       date_timestamp= ? ORDER BY date_timestamp DESC";
12       $stat = $connection->prepare($sql);
13       // prepare() can fail because of syntax errors, missing privileges,
       ....
14       if (false===$stat)
15       {
16           die('prepare() failed: ' . htmlspecialchars($mysqli->error));
17       }
18
19       $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
20       $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
21
22       $result = $stat->execute([$VAT_doctor_SELECTED,
       $date_timestamp_SELECTED]);
23       if ($result == FALSE)
24       {
25         $info = $connection->errorInfo();
26         echo("<p>Error: {$info[2]}</p>");
27         exit();
28       }
29
30       echo("<table border=\"1\">");
31       echo("<tr><td>VAT Doctor</td><td>Date</td><td>Description</td><td>
       VAT Client</td></tr>");
32       $result = $stat->fetchAll();
33       foreach($result as $row)
34       {
35         echo("<tr><td>");
36         echo($row['VAT_doctor']);
```

```
37        echo("</td><td>");
38        echo($row['date_timestamp']);
39        echo("</td><td>");
40        echo($row['description_']);
41        echo("</td><td>");
42        echo($row['VAT_client']);
43        echo("</td></tr>");
44      }
45      echo("</table>");
46       $connection = null;
47    ?>
48    <form action="appointment_info.php" method="post" VAT_client =
      VAT_client>
49      <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
      VAT_client']?>">
50        <p><input type="submit" value="Back to Appointment/Consultation
      list"/></p>
51    </form>
52    </body>
53 </html>
```

To see all the info related to a specific consultation the user must select the reference "Get Info" on the Consultation list. Once selected this page is redirected to the page "info_consultation.php" which receives the VAT_client and the date_time_stamp and presents the information presented in figure 11

Information about Consultation with Doctor 88888888 on Date 2019-12-06 17:00:00

General Information:

VAT Doctor	Data	SOAP S	SOAP O	SOAP A	SOAP P
88888888	2019-12-06 17:00:00	z	He has gingivitis	x	y

Diagnostic Info:

Diagnostic Code	Diagnostic Description
980000006	dental cavities, swelling of the face and cheek

Prescription Info:

Diagnostic Code	Medication Name	Laboratory	Dosage	Description
980000006	Valium	Servier Laboratories	Three doses	Every five hours, for 7 week

Assistant Info:

VAT Nurse
54445867

Procedures:

Name	Type	Edit Info
molar extraction	tooth extractions	Cant edit
upper side map chart	dental charting	Edit Info

Back to Appointment/Consultation list

Figure 11: Consultation Info page

The code for this page is more complex since a lot of information is presented (SOAP notes, Diagnostic Codes, Prescription Info, Assistant Info and Procedures). The source code for this page is the following:

```php
1  <html>
2    <head>
3      <title>Consultation Info</title>
4    </head>
5     <body>
6     <h3>Information about Consultation with Doctor <?=$_REQUEST['
       VAT_doctor']?> on Date <?=$_REQUEST['date_timestamp']?></h3>
7     <h4>General Information: </h4>
8     <?php
9       include 'connect_db.php';
10
11      $sql = "SELECT c.VAT_doctor, c.date_timestamp, c.SOAP_S, c.SOAP_O,
       c.SOAP_A, c.SOAP_P
12      FROM consultation as c WHERE c.VAT_doctor= ?
13      AND c.date_timestamp=?";
14
15      $stat = $connection->prepare($sql);
16      // prepare() can fail because of syntax errors, missing privileges,
       ....
17      if (false===$stat)
18      {
19         die('prepare() failed: ' . htmlspecialchars($mysqli->error));
20      }
21
22      $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
23      $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
24
25      $result = $stat->execute([$VAT_doctor_SELECTED,
     $date_timestamp_SELECTED]);
26      if ($result == FALSE)
27      {
28        $info = $connection->errorInfo();
29        echo("<p>Error: {$info[2]}</p>");
30        exit();
31      }
32
33      echo("<table border=\"1\">");
34      echo("<tr><td>VAT Doctor</td><td>Data</td><td>SOAP S</td><td>SOAP O
```

```
   </td><td>SOAP A</td><td>SOAP P</td></tr>");
35   $result = $stat->fetchAll();
36   foreach($result as $row)
37   {
38     echo("<tr><td>");
39     echo($row['VAT_doctor']);
40     echo("</td><td>");
41     echo($row['date_timestamp']);
42     echo("</td><td>");
43     echo($row['SOAP_S']);
44     echo("</td><td>");
45     echo($row['SOAP_O']);
46     echo("</td><td>");
47     echo($row['SOAP_A']);
48     echo("</td><td>");
49     echo($row['SOAP_P']);
50     echo("</td></tr>");
51   }
52   echo("</table>");
53    $connection = null;
54 ?>
55 <h4>Diagnostic Info: </h4>
56 <?php
57   include 'connect_db.php';
58
59   $sql = "SELECT cd.ID, dc.description
60   FROM consultation as c
61   LEFT JOIN consultation_diagnostic as cd ON c.VAT_doctor = cd.
   VAT_doctor AND c.date_timestamp = cd.date_timestamp
62   LEFT JOIN  diagnostic_code  as dc ON cd.ID = dc.ID
63   WHERE c.VAT_doctor= ? AND c.date_timestamp= ? ";
64
65   $stat = $connection->prepare($sql);
66   // prepare() can fail because of syntax errors, missing privileges,
   ....
67   if (false===$stat)
68   {
69       die('prepare() failed: ' . htmlspecialchars($mysqli->error));
70   }
71
72   $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
73   $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
```

```php
74
75      $result = $stat->execute([$VAT_doctor_SELECTED,
    $date_timestamp_SELECTED]);
76      if ($result == FALSE)
77      {
78        $info = $connection->errorInfo();
79        echo("<p>Error: {$info[2]}</p>");
80        exit();
81      }
82
83      echo("<table border=\"1\">");
84      echo("<tr><td>Diagnostic Code</td><td>Diagnostic Description</td></
    tr>");
85      $result = $stat->fetchAll();
86      foreach($result as $row)
87      {
88        echo("<tr><td>");
89        echo($row['ID']);
90        echo("</td><td>");
91        echo($row['description']);
92        echo("</td></tr>");
93      }
94      echo("</table>");
95       $connection = null;
96    ?>
97    <h4>Prescription Info: </h4>
98    <?php
99      include 'connect_db.php';
100
101     $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
102     $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
103
104     $sql = "SELECT cd.ID, p.name_, p.lab, p.dosage, p.description_
105     FROM consultation as c
106     LEFT JOIN consultation_diagnostic as cd ON c.VAT_doctor = cd.
    VAT_doctor AND c.date_timestamp = cd.date_timestamp
107     LEFT JOIN prescription as p ON c.VAT_doctor = p.VAT_doctor AND c.
    date_timestamp = p.date_timestamp AND cd.ID = p.ID
108     WHERE c.VAT_doctor= ? AND c.date_timestamp= ?";
109     $stat = $connection->prepare($sql);
110     // prepare() can fail because of syntax errors, missing privileges,
    ....
```

```php
111      if (false===$stat)
112      {
113          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
114      }
115
116      $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
117      $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
118
119      $result = $stat->execute([$VAT_doctor_SELECTED,
     $date_timestamp_SELECTED]);
120      if ($result == FALSE)
121      {
122        $info = $connection->errorInfo();
123        echo("<p>Error: {$info[2]}</p>");
124        exit();
125      }
126
127      echo("<table border=\"1\">");
128      echo("<tr><td>Diagnostic Code</td><td>Medication Name</td><td>
     Laboratory</td><td>Dosage</td><td>Description</td></tr>");
129      $result = $stat->fetchAll();
130      foreach($result as $row)
131      {
132        echo("<tr><td>");
133        echo($row['ID']);
134        echo("</td><td>");
135        echo($row['name_']);
136        echo("</td><td>");
137        echo($row['lab']);
138        echo("</td><td>");
139        echo($row['dosage']);
140        echo("</td><td>");
141        echo($row['description_']);
142        echo("</td></tr>");
143      }
144      echo("</table>");
145       $connection = null;
146    ?>
147    <h4>Assistant Info: </h4>
148    <?php
149      include 'connect_db.php';
150
```

```php
151    $sql = "SELECT cs.VAT_nurse
152    FROM consultation as c
153    LEFT JOIN consultation_assistant as cs ON cs.VAT_doctor = c.
       VAT_doctor AND cs.date_timestamp = c.date_timestamp
154    WHERE c.VAT_doctor= ? AND c.date_timestamp= ?";
155    $stat = $connection->prepare($sql);
156    // prepare() can fail because of syntax errors, missing privileges,
       ....
157    if (false===$stat)
158    {
159        die('prepare() failed: ' . htmlspecialchars($mysqli->error));
160    }
161
162    $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
163    $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
164
165    $result = $stat->execute([$VAT_doctor_SELECTED,
       $date_timestamp_SELECTED]);
166    if ($result == FALSE)
167    {
168      $info = $connection->errorInfo();
169      echo("<p>Error: {$info[2]}</p>");
170      exit();
171    }
172
173    echo("<table border=\"1\">");
174    echo("<tr><td>VAT Nurse</td></tr>");
175    $result = $stat->fetchAll();
176    foreach($result as $row)
177    {
178      echo("<tr><td>");
179      echo($row['VAT_nurse']);
180      echo("</td></tr>");
181    }
182    echo("</table>");
183     $connection = null;
184    ?>
185    <h4>Procedures: </h4>
186    <?php
187    include 'connect_db.php';
188
189    $sql = "SELECT p.name_, p.type_
```

```
190        FROM consultation as c
191        LEFT JOIN procedure_in_consultation as pc ON pc.VAT_doctor = c.
       VAT_doctor AND pc.date_timestamp = c.date_timestamp
192        LEFT JOIN procedure_ as p ON pc.name_ = p.name_
193        WHERE c.VAT_doctor= ? AND c.date_timestamp= ?";
194        $stat = $connection->prepare($sql);
195        // prepare() can fail because of syntax errors, missing privileges,
       ....
196        if (false===$stat)
197        {
198           die('prepare() failed: ' . htmlspecialchars($mysqli->error));
199        }
200
201        $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
202        $date_timestamp_SELECTED = $_REQUEST['date_timestamp'];
203        $VAT_client = $_REQUEST['VAT_client'];
204
205        $result = $stat->execute([$VAT_doctor_SELECTED,
       $date_timestamp_SELECTED]);
206        if ($result == FALSE)
207        {
208          $info = $connection->errorInfo();
209          echo("<p>Error: {$info[2]}</p>");
210          exit();
211        }
212
213        echo("<table border=\"1\">");
214        echo("<tr><td>Name </td><td>Type </td><td>Edit Info </td></tr>");
215        $result = $stat->fetchAll();
216        foreach($result as $row)
217        {
218          echo("<tr><td>");
219          echo($row['name_']);
220          echo('</td><td>');
221          echo($row['type_']);
222          if($row['type_']=='dental charting'){
223            echo('</td><td>');
224            echo("<a href=\"add_info_consultation_charting.php?VAT_doctor="
       . $VAT_doctor_SELECTED . "&date_timestamp=" .
       $date_timestamp_SELECTED . "&name_=" . $row['name_'] . "&VAT_client="
       . $VAT_client);
225            echo("\">Edit Info </a></td>\n");
```

```
226        }
227      else{
228        echo('</td><td>');
229        echo('Cant edit');
230      }
231      echo("</td></tr>");
232    }
233    echo("</table>");
234     $connection = null;
235  ?>
236  <form action="appointment_info.php" method="post" VAT_client =
     VAT_client>
237    <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
     VAT_client']?>">
238      <p><input type="submit" value="Back to Appointment/Consultation
     list"/></p>
239  </form>
240  </body>
241 </html>
```

Besides accessing all the information, the registry of new information is also supported. By selecting the reference "+ADD info" in the appointments list the user is allowed to add information to that particular appointment and convert it into a consultation. The reference "+ADD info" redirects the page to the page "add_info_consultation_pre.php" in which the VAT_doctor and date_time_stamp are received. This page is presented in the following figure 12.

Creating a new consultation with Doctor 88888888 on Date 2019-02-05 17:00:00

Please Add Soap Notes in order to create a Consultation

Add SOAP Notes:

SOAP S: This

SOAP O: Is

SOAP A: A

SOAP P: Test

Add SOAP Note

Figure 12: Adding SOAP notes page

All this input fields are required which means that they are mandatory. The source code is the following:

```
1 <html>
2   <head>
3     <title>Add Appointment Info</title>
```

```
4   </head>
5   <h3>Creating a new consultation with Doctor <?=$_REQUEST['VAT_doctor'
    ]?> on Date <?=$_REQUEST['date_timestamp']?></h3>
6   <h4>Please Add Soap Notes in order to create a Consultation </h4>
7    <body>
8       <h4> Add SOAP Notes: </h4>
9       <form action="update_info_consultation_SOAP.php" method="post"
10          VAT_doctor = VAT_doctor Date=Date VAT_client = VAT_client
11          SOAP_S=SOAP_S SOAP_O=SOAP_O SOAP_A=SOAP_A SOAP_P=SOAP_P>
12          <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
13          <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
    VAT_doctor']?>">
14          <input type="hidden" name="Date" value="<?=$_REQUEST['
    date_timestamp']?>">
15          <p>SOAP S: <input type="text" name="SOAP_S" required/></p>
16          <p>SOAP O: <input type="text" name="SOAP_O" required/></p>
17          <p>SOAP A: <input type="text" name="SOAP_A" required/></p>
18          <p>SOAP P: <input type="text" name="SOAP_P" required/></p>
19          <p><input type="submit" value="Add SOAP Note"/></p>
20       </form>
21       </form>
22    </body>
23 </html>
```

Once all the SOAP Notes info is inserted, the action "Add SOAP Note" redirects the page to "update_info_info_consultation_SOAP.php" which receives the VAT_doctor, date_time_stamp, SOAP_S, SOAP_S, SOAP_S,SOAP_S, and creates a new consultation. Once this page is created, the user can go back to the list of appointments/consultations for that client or add more info regarding the nurses, diagnostics and prescriptions. This intermediary step of adding the soap notes and creating a consultation first, ensures that when adding information related to Nurses, Diagnostics And Prescriptions the consultation is already created.

This page is presented in the following Figure 13.

Information Was Updated! Add Nurses/Diagnostic/Percription More Info:

Add Nurses/Diagnostic/Percription

Atention! Once you go back, you cant add more info!:

Back to Appointment/Consultation list

Figure 13: Update SOAP and Go Back to Appointment and Consultations List

The code for this page is the following:

```
1  <html>
2   <body>
3     <?php
4       include 'connect_db.php';
5
6       $sql = "INSERT INTO consultation VALUES(?,?,?,?,?,?);'";
7       $stat = $connection->prepare($sql);
8       // prepare() can fail because of syntax errors, missing privileges
   , ....
9       if (false===$stat)
10      {
11         die('prepare() failed: ' . htmlspecialchars($mysqli->error));
12      }
13
14      $VAT_doctor_UPDATE = $_REQUEST['VAT_doctor'];
15      $Date_UPDATE = $_REQUEST['Date'];
16      $SOAP_S_UPDATE = $_REQUEST['SOAP_S'];
17      $SOAP_O_UPDATE = $_REQUEST['SOAP_O'];
18      $SOAP_A_UPDATE = $_REQUEST['SOAP_A'];
19      $SOAP_P_UPDATE = $_REQUEST['SOAP_P'];
20
21      $result = $stat->execute([$VAT_doctor_UPDATE, $Date_UPDATE,
   $SOAP_S_UPDATE, $SOAP_O_UPDATE, $SOAP_A_UPDATE, $SOAP_P_UPDATE]);
22      if ($result == FALSE)
23      {
24        $info = $connection->errorInfo();
25        echo("<p>Error: {$info[2]}</p>");
26        exit();
27      }
28
29      $connection = null;
30    ?>
31    <h4>Information Was Updated! Add Nurses/Diagnostic/Percription More
   Info: </h4>
32    <form action="add_info_consultation.php" method="post" VAT_doctor=
   VAT_doctor Date=Date VAT_client=VAT_client>
33      <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
   VAT_client']?>">
34      <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
   VAT_doctor']?>">
35      <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>">
```

```
36        <p><input type="submit" value="Add Nurses/Diagnostic/Percription
    "/></p>
37     </form>
38     <h4>Atention! Once you go back, you cant add more info!: </h4>
39     <form action="appointment_info.php" method="post" VAT_client=
    VAT_client>
40         <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
41         <p><input type="submit" value="Back to Appointment/Consultation
    list"/></p>
42     </form>
43  </body>
44 </html>
```

The action "Add Nurses/Diagnostic/Prescription" redirects the page to "add_info _consultation.php" which allows the user to add more information to that consultation. In this page Several Nurses, Diagnostics and Prescriptions can be added. The page is represented in the following image 14.

Figure 14: Add Nurse/Diagnostic/Prescription to Consultation

All the parameters in this form are required. The list of available nurses only presents the nurses available at that time. Besides, the list of Diagnostic codes only present the ones that have not been added to that consultation yet, avoiding duplicate information. It is only possible to add a prescription for a diagnostic code, if the consultation has none the user needs to add a diagnostic code first in order to have a prescription. The source code is the following:

```
1 <html>
2   <head>
```

```php
3      <title>Add Appointment Info</title>
4    </head>
5    <h3>Insert More Info into with Doctor <?=$_REQUEST['VAT_doctor']?> on
     Date <?=$_REQUEST['Date']?></h3>
6    <body>
7      <h4> Add Nurse: </h4>
8      <form action="update_info_consultation_nurse.php" method="post"
9        VAT_doctor = VAT_doctor Date=Date VAT_nurse = VAT_nurse
     VAT_client=VAT_client>
10         <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
     VAT_client']?>">
11         <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
     VAT_doctor']?>">
12         <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>">
13         <p>Available VAT Nurse:
14           <select name="VAT_nurse" required>
15           <?php
16             include 'connect_db.php';
17
18             /*Sees the available nurses at that time*/
19             $sql = "SELECT DISTINCT VAT_nurse FROM
     consultation_assistant
20             WHERE VAT_nurse NOT IN (SELECT DISTINCT VAT_nurse FROM
     consultation_assistant as cs
21             WHERE cs.date_timestamp= ?);";
22             $stat = $connection->prepare($sql);
23             // prepare() can fail because of syntax errors, missing
     privileges, ....
24             if (false===$stat)
25             {
26                 die('prepare() failed: ' . htmlspecialchars($mysqli->
     error));
27             }
28
29             $date_timestamp_SELECTED = $_REQUEST['Date'];
30
31             $result = $stat->execute([$date_timestamp_SELECTED]);
32             if ($result == FALSE)
33             {
34               $info = $connection->errorInfo();
35               echo("<p>Error: {$info[2]}</p>");
36               exit();
```

```php
37              }

39              echo("<table border=\"1\">");
40              $result = $stat->fetchAll();
41              foreach($result as $row)
42              {
43                  $VAT_nurse = $row['VAT_nurse'];
44                  echo("<option value=\"$VAT_nurse\">$VAT_nurse</option>");
45               }
46               $connection = null;
47          ?>
48         </select>
49      </p>
50       <p><input type="submit" value="Add Nurse"/></p>
51      </form>
52      <h4> Add Diagnostic: </h4>
53      <form action="update_info_consultation_diagnostic.php" method="
    post"
54          VAT_doctor = VAT_doctor Date=Date VAT_client=VAT_client
55          Diagnostic_ID=Diagnostic_ID>
56          <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
57          <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
    VAT_doctor']?>">
58          <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>"
    >
59          <p>Availabe Diagnostic ID and Description:
60            <select name="Diagnostic_ID" required>
61            <?php
62              include 'connect_db.php';

64              /*Selects the remaining diagnostic codes*/
65              $sql = "SELECT DISTINCT ID, description FROM
    diagnostic_code
66              WHERE ID NOT IN (SELECT DISTINCT ID FROM
    consultation_diagnostic as cd
67              WHERE cd.date_timestamp= ? AND cd.VAT_doctor = ? );";
68              $stat = $connection->prepare($sql);
69              // prepare() can fail because of syntax errors, missing
    privileges, ....
70              if (false===$stat)
71              {
```

```php
72            die('prepare() failed: ' . htmlspecialchars($mysqli->
   error));
73          }
74
75          $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
76          $date_timestamp_SELECTED = $_REQUEST['Date'];
77
78          $result = $stat->execute([$date_timestamp_SELECTED,
   $VAT_doctor_SELECTED]);
79          if ($result == FALSE)
80          {
81            $info = $connection->errorInfo();
82            echo("<p>Error: {$info[2]}</p>");
83            exit();
84          }
85
86          $result = $stat->fetchAll();
87          foreach($result as $row)
88          {
89            $Diagnostic_ID = $row['ID'];
90            $Description = $row['ID'].", ".$row['description'];
91            echo("<option value=\"$Diagnostic_ID\">$Description</
   option>");
92          }
93          $connection = null;
94        ?>
95        </select>
96      </p>
97      <p><input type="submit" value="Add Diagnostic Code"/></p>
98    </form>
99    <h4> Add Prescription: </h4>
100   <form action="update_info_consultation_prescription.php" method="
   post"
101      VAT_doctor = VAT_doctor Date=Date ID_diagnostic_prescription=
   ID_diagnostic_prescription VAT_client=VAT_client
102      Medication_Name=Medication_Name dosage=dosage description=
   description>
103      <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
   VAT_client']?>">
104      <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
   VAT_doctor']?>">
105      <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>"
```

```php
    >
        <p>Available Diagnostic Code:
            <select name="ID_diagnostic_prescription" required>
            <?php
                include 'connect_db.php';

                /*Selects the remaining diagnostic codes*/
                $sql = "SELECT DISTINCT ID FROM consultation_diagnostic as cd
                WHERE cd.date_timestamp='$date_timestamp_SELECTED'
                AND cd.VAT_doctor = '$VAT_doctor_SELECTED'
                AND ID NOT IN (SELECT DISTINCT ID FROM prescription as p
                WHERE p.date_timestamp= ? AND p.VAT_doctor = ? );";
                $stat = $connection->prepare($sql);
                // prepare() can fail because of syntax errors, missing privileges, ....
                if (false===$stat)
                {
                    die('prepare() failed: ' . htmlspecialchars($mysqli->error));
                }

                $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
                $date_timestamp_SELECTED = $_REQUEST['Date'];

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

                echo("<table border=\"1\">");
                $result = $stat->fetchAll();
                foreach($result as $row)
                {
                    $ID_diagnostic_prescription = $row['ID'];
                    echo("<option value=\"$ID_diagnostic_prescription\">$ID_diagnostic_prescription</option>");
                }
```

```php
142              $connection = null;
143            ?>
144          </select>
145        </p>
146         <p>Prescription Info:
147           <select name="Medication_Name" required>
148           <?php
149             include 'connect_db.php';
150
151             $sql = "SELECT * FROM medication;";
152             $stat = $connection->prepare($sql);
153             // prepare() can fail because of syntax errors, missing
    privileges, ....
154             if (false===$stat)
155             {
156                 die('prepare() failed: ' . htmlspecialchars($mysqli->
    error));
157             }
158
159             $VAT_doctor_SELECTED = $_REQUEST['VAT_doctor'];
160             $date_timestamp_SELECTED = $_REQUEST['Date'];
161
162             $result = $stat->execute();
163             if ($result == FALSE)
164             {
165               $info = $connection->errorInfo();
166               echo("<p>Error: {$info[2]}</p>");
167               exit();
168             }
169
170             echo("<table border=\"1\">");
171             $result = $stat->fetchAll();
172             foreach($result as $row)
173             {
174                 $name = $row['name_'];
175                 $Medication_Lab = $row['lab'];
176                 $Medication_Name = $name.", ".$Medication_Lab;
177                 echo("<option value=\"$Medication_Name\">
    $Medication_Name</option>");
178               }
179               $connection = null;
180            ?>
```

```
181              </select>
182           </p>
183           <p>Dosage : <input type="text" name="dosage" required/></p>
184           <p>Description : <input type="text" name="description" required
      /></p>
185           <p><input type="submit" value="Add Prescription"/></p>
186        </form>
187        <h4>Atention! Once you go back, you cant add more info!: </h4>
188        <form action="appointment_info.php" method="post" VAT_client=
      VAT_client>
189           <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
      VAT_client']?>">
190              <p><input type="submit" value="Back to Appointment/
      Consultation list"/></p>
191        </form>
192     </body>
193 </html>
```

The actions "Add Nurse", "Add Diagnostic Code" and "Add Prescription" redirect
to the pages "update_info_consultation_nurse.php" "update_info_consultation _diag-
nostic.php" and "update_info_consultation_prescription.php", respectively. This pages
are very similar, they both receive the information to be added and add it to the data
base. All these pages have the action "Add More Info" which redirects the page to the
previous one for adding more Nurses/Diagnostics/Prescriptions and the "Back to Ap-
pointment/Consultation list" which goes back to the list of appointments/consultations
for that client.

The pages for updating the Nurse, Prescription and Diagnostic information are
presented in figure 15.

Add more Information to Consultation:

Add More Info

Information Was Updated! Finish Adding Information:

Back to Appointment/Consultation list

Figure 15: Updated Nurse/Diagnostic/Prescription to Information

The code for "update_info_consultation_nurse.php" is the following:

```
1 <html>
2  <body>
```

```php
 3    <?php
 4      include 'connect_db.php';
 5
 6      $sql = "INSERT INTO consultation_assistant VALUES(?,?,?);'";
 7      $stat = $connection->prepare($sql);
 8      // prepare() can fail because of syntax errors, missing privileges
    , ....
 9      if (false===$stat)
10      {
11          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
12      }
13
14      $VAT_doctor_UPDATE = $_REQUEST['VAT_doctor'];
15      $Date_UPDATE = $_REQUEST['Date'];
16      $VAT_nurse_UPDATE = $_REQUEST['VAT_nurse'];
17
18      $result = $stat->execute([$VAT_doctor_UPDATE, $Date_UPDATE,
    $VAT_nurse_UPDATE]);
19      if ($result == FALSE)
20      {
21        $info = $connection->errorInfo();
22        echo("<p>Error: {$info[2]}</p>");
23        exit();
24      }
25
26      $connection = null;
27    ?>
28    <h4>Add more Information to Consultation: </h4>
29    <form action="add_info_consultation.php" method="post" VAT_doctor=
    VAT_doctor Date=Date VAT_client=VAT_client>
30      <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
31      <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
    VAT_doctor']?>">
32      <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>">
33        <p><input type="submit" value="Add More Info"/></p>
34    </form>
35    <h4>Information Was Updated! Finish Adding Information: </h4>
36    <form action="appointment_info.php" method="post" VAT_client=
    VAT_client>
37      <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
```

```
38          <p><input type="submit" value="Back to Appointment/Consultation
    list"/></p>
39      </form>
40   </body>
41 </html>
```

The code for "update_info_consultation_diagnostic.php" is the following:

```
1 <html>
2  <body>
3     <?php
4        include 'connect_db.php';
5
6        $sql = "INSERT INTO consultation_diagnostic VALUES(?,?,?);'";
7        $stat = $connection->prepare($sql);
8        // prepare() can fail because of syntax errors, missing privileges
    , ....
9        if (false===$stat)
10       {
11          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
12       }
13
14       $VAT_doctor_UPDATE = $_REQUEST['VAT_doctor'];
15       $Date_UPDATE = $_REQUEST['Date'];
16       $Diagnostic_ID_UPDATE = $_REQUEST['Diagnostic_ID'];
17
18       $result = $stat->execute([$VAT_doctor_UPDATE, $Date_UPDATE,
    $Diagnostic_ID_UPDATE]);
19       if ($result == FALSE)
20       {
21         $info = $connection->errorInfo();
22         echo("<p>Error: {$info[2]}</p>");
23         exit();
24       }
25
26       $connection = null;
27    ?>
28    <h4>Add more Information to Consultation: </h4>
29    <form action="add_info_consultation.php" method="post" VAT_doctor=
    VAT_doctor Date=Date VAT_client=VAT_client>
30       <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
31       <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
```

```
     VAT_doctor ']? > " >
32      < input type = " hidden " name = " Date " value = " <?= $_REQUEST [ ' Date ']? > " >
33        < p > < input type = " submit " value = " Add More Info " / > </ p >
34    </ form >
35    < h4 > Information Was Updated ! Finish Adding Information : </ h4 >
36    < form action = " appointment_info.php " method = " post " VAT_client =
     VAT_client >
37      < input type = " hidden " name = " VAT_client " value = " <?= $_REQUEST [ '
     VAT_client ']? > " >
38        < p > < input type = " submit " value = " Back to Appointment / Consultation
     list " / > </ p >
39    </ form >
40  </ body >
41 </ html >
```

The code for "update_info_consultation_prescription.php" is the following:

```
1 < html >
2  < body >
3    <? php
4      include ' connect_db.php ';
5
6      $sql = " INSERT INTO prescription VALUES (?, ?, ?, ?, ?, ?, ?) ;' ";
7      $stat = $connection - > prepare ( $sql );
8      // prepare () can fail because of syntax errors , missing privileges
     , ....
9      if ( false === $stat )
10     {
11       die ( ' prepare () failed : ' . htmlspecialchars ( $mysqli - > error ));
12     }
13
14     $VAT_doctor_UPDATE = $_REQUEST [ ' VAT_doctor '];
15     $Date_UPDATE = $_REQUEST [ ' Date '];
16     $ID_diagnostic_prescription = $_REQUEST [ '
     ID_diagnostic_prescription '];
17     $name_UPDATE = $_REQUEST [ ' Medication_Name '];
18     $medication = explode ( ", ", $name_UPDATE );
19     $dosage_UPDATE = $_REQUEST [ ' dosage '];
20     $description_UPDATE = $_REQUEST [ ' description '];
21
22     $result = $stat - > execute ([ $medication [0] , $medication [1] ,
     $VAT_doctor_UPDATE , $Date_UPDATE , $ID_diagnostic_prescription ,
23      $dosage_UPDATE , $description_UPDATE ]);
```

```
24        if ($result == FALSE)
25        {
26          $info = $connection->errorInfo();
27          echo("<p>Error: {$info[2]}</p>");
28          exit();
29        }
30
31        $connection = null;
32     ?>
33     <h4>Add more Information to Consultation: </h4>
34     <form action="add_info_consultation.php" method="post" VAT_doctor=
       VAT_doctor Date=Date VAT_client=VAT_client>
35        <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
       VAT_client']?>">
36        <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
       VAT_doctor']?>">
37        <input type="hidden" name="Date" value="<?=$_REQUEST['Date']?>">
38          <p><input type="submit" value="Add More Info"/></p>
39     </form>
40     <h4>Information Was Updated! Finish Adding Information: </h4>
41     <form action="appointment_info.php" method="post" VAT_client=
       VAT_client>
42        <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
       VAT_client']?>">
43          <p><input type="submit" value="Back to Appointment/Consultation
       list"/></p>
44     </form>
45  </body>
46 </html>
```

1.3   3rd Task

In this Task a group of pages were created in order to update a dental charting associated to a specific consultation. All the consultations that have this particular type of procedure, have an option "Edit Info" in the procedures table located in the page that has all the detailed information about the consultation (figure 11). The following figure 16 shows an example of a consultation where editing a dental charting is possible.

Procedures:

Name	Type	Edit Info
molar extraction	tooth extractions	Cant edit
upper side map chart	dental charting	Edit Info

Back to Appointment/Consultation list

Figure 16: Edit Info Dental Charting

This reference redirects the page to "add_info_consultation_charting.php". This page is a form where the list of all the teeth is presented followed by a description and measure input fields. In this page the user can add info for each teeth and it is presented in figure 17

Dental Charting Form :

Quadrant	Number	Name	Description	Measure
Lower Left	17	Wisdom Tooth (3rd Molar)		
Lower Left	18	Molar (2nd Molar)		
Lower Left	19	Molar (1st Molar)		
Lower Left	20	Bicuspid (2nd)		
Lower Left	21	Bicuspid (1st)		
Lower Left	22	Canine (Eye tooth / Cuspid)		
Lower Left	23	Incisor (Lateral)		
Lower Left	24	Incisor (Central)		
Lower Right	25	Incisor (Central)		
Lower Right	26	Incisor (Lateral)		
Lower Right	27	Canine (Eye tooth / Cuspid)		
Lower Right	28	Bicuspid (1st)		
Lower Right	29	Bicuspid (2nd)		
Lower Right	30	Molar (1st Molar)		
Lower Right	31	Molar (2nd Molar)		
Lower Right	32	Wisdom Tooth (3rd Molar)		
Upper Left	9	Incisor (Central)		
Upper Left	10	Incisor (Lateral)		
Upper Left	11	Canine (Eye tooth / Cuspid)		
Upper Left	12	Bicuspid (1st)		
Upper Left	13	Bicuspid (2nd)		
Upper Left	14	Molar (1st)		
Upper Left	15	Molar (2nd)		
Upper Left	16	Wisdom Tooth (3rd Molar)		
Upper Right	1	Wisdom Tooth (3rd Molar)		
Upper Right	2	Molar (2nd Molar)		
Upper Right	3	Molar (1st Molar)		
Upper Right	4	Bicuspid (2nd)		
Upper Right	5	Bicuspid (1st)		
Upper Right	6	Canine (Eye tooth / Cuspid)		
Upper Right	7	Incisor (Lateral)		
Upper Right	8	Incisor (Central)		

Insert Dental Charting

Figure 17: Add Info Dental Charting

In this page the inputs are not required since not all teeth need to be measured. The source code for this page is the following:

```
1  <html>
2   <body>
```

```php
3    <form action="update_info_consultation_dcharting.php" method="post"
4     VAT_doctor=VAT_doctor Date=Date name_=name_ VAT_client=VAT_client
5     Description=Description Measure=Measure>
6       <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
7       <input type="hidden" name="VAT_doctor" value="<?=$_REQUEST['
    VAT_doctor']?>">
8       <input type="hidden" name="Date" value="<?=$_REQUEST['
    date_timestamp']?>">
9       <input type="hidden" name="name_" value="<?=$_REQUEST['name_']?>">
10   <h4>Dental Charting Form : </h4>
11   <?php
12      include 'connect_db.php';
13
14      $sql = "SELECT * FROM teeth;";
15      $stat = $connection->prepare($sql);
16      // prepare() can fail because of syntax errors, missing privileges
    , ....
17      if (false===$stat)
18      {
19         die('prepare() failed: ' . htmlspecialchars($mysqli->error));
20      }
21
22      $result = $stat->execute();
23      if ($result == FALSE)
24      {
25        $info = $connection->errorInfo();
26        echo("<p>Error: {$info[2]}</p>");
27        exit();
28      }
29      echo("<table border=\"1\">");
30      echo("<tr><td>Quadrant</td><td>Number</td><td>Name</td><td>
    Description</td><td>Measure</td></tr>");
31      $result = $stat->fetchAll();
32      foreach($result as $row)
33      {
34        echo("<tr><td>");
35        echo($row['quadrant']);
36        echo("</td><td>");
37        echo($row['number_']);
38        echo("</td><td>");
39        echo($row['name_']);
```

```php
40        echo("</td><td>");
41        ?>
42          <p><input type="text" name="Description[]"/></p>
43        <?php echo("</td><td>");?>
44          <p><input type="text" name="Measure[]"/></p>
45        <?php
46        echo("</td></tr>");
47      }
48      echo("</table>");
49       $connection = null;
50    ?>
51      <p><input type="submit" value="Insert Dental Charting"/></p>
52    </form>
53    <h4>Go back to list: </h4>
54    <form action="appointment_info.php" method="post" VAT_client =
    VAT_client >
55        <p><input type="submit" value="Back to Appointment/Consultation
    list"/></p>
56        <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
    VAT_client']?>">
57    </form>
58  </body>
59 </html>
```

Once the form is completed the user can "Insert Dental Charting" and the page will be redirected to "update_info_consultation_dcharting.php". In this page, the information for each tooth is received and updated into the the database. If the user does not insert any description or measure, the added description and measure will be "No Description Added" and 0.00, respectively. Then the user can return to the list of all appointments and consultations, figure 18.

Information Was Updated! Go back to list:

Back to Appointment/Consultation list

Figure 18: Updated Info Dental Charting

The infomation is updated using a single transaction. The source code is the following:

```html
1 <html>
2  <body>
3    <?php
```

```php
4       include 'connect_db.php';
5       $index = 0;
6
7      $sql = "SELECT * FROM teeth;";
8       $stat = $connection ->prepare($sql);
9       // prepare() can fail because of syntax errors, missing privileges,
        ....
10      if (false===$stat)
11      {
12         die('prepare() failed: ' . htmlspecialchars($mysqli ->error));
13      }
14
15      $result = $stat ->execute();
16      if ($result == FALSE)
17      {
18        $info = $connection ->errorInfo();
19        echo("<p>Error: {$info[2]}</p>");
20        exit();
21      }
22
23      $result = $stat ->fetchAll();
24      foreach($result as $row)
25      {
26       $quadrant[$index] = $row['quadrant'];
27       $number_[$index] = $row['number_'];
28       $name_[$index] = $row['name_'];
29       $index=$index+1;
30     }
31      $connection = null;
32    ?>
33     <?php
34      include 'connect_db.php';
35
36      $VAT_doctor_UPDATE = $_REQUEST['VAT_doctor'];
37      $Date_UPDATE = $_REQUEST['Date'];
38      $measure = $_REQUEST['Measure'];
39      $description = $_REQUEST['Description'];
40      $name_ = $_REQUEST['name_'];
41
42      //Initiates transaction
43      $connection ->beginTransaction();
44
```

```
45      for($i=0; $i < sizeof($measure); $i++){
46        if(empty($measure[$i])){
47          $measure_insert = 0;
48        }
49        else{
50          $measure_insert = $measure[$i];
51        }
52        if(empty($description[$i])){
53          $description_insert = 'No Description Added';
54        }
55        else{
56          $description_insert = $description[$i];
57        }
58
59        $sql = "INSERT INTO procedure_charting VALUES (?,?,?,?,?,?,?);";
60        $stat = $connection->prepare($sql);
61        // prepare() can fail because of syntax errors, missing
      privileges, ....
62        if (false===$stat)
63        {
64          die('prepare() failed: ' . htmlspecialchars($mysqli->error));
65        }
66
67        $stat->execute([$name_,$VAT_doctor_UPDATE,$Date_UPDATE,$quadrant
      [$i],$number_[$i],$description_insert,$measure_insert]);
68      }
69
70      //Executes the transaction
71      $result = $connection->commit();
72      //Checks if error exists
73      if ($result == FALSE)
74      {
75        $info = $connection->errorInfo();
76        echo("<p>Error: {$info[2]}</p>");
77        exit();
78      }
79      $connection = null;
80    ?>
81     <h4>Information Was Updated! Go back to list: </h4>
82      <form action="appointment_info.php" method="post" VAT_client=
      VAT_client>
83        <input type="hidden" name="VAT_client" value="<?=$_REQUEST['
```

```
        VAT_client']?>">
84          <p><input type="submit" value="Back to Appointment/Consultation
        list"/></p>
85        </form>
86    </body>
87  </html>
```

1.4   URL with the Web application

The URL to our web application is the following:

http://web.ist.utl.pt/ist187024/search_clients.php

or

http://web.ist.utl.pt/ist425312/search_clients.php

Both url have the same web application and content.

1.4.1   Brief instructions on how to use the webpage

The landing page of our webpage is the search for the clients. After searching the clients, appears for each of them a link to create a new appointment and to show a list of appointments/consultations for that client.

If you desire to create a new appointment, it takes it to a page to select a date and a time for the appointment. After you select a date and a time for the appointment, it will direct you to a page where you can select a doctor from all available. At each of the doctors, there is a link 'Create Appointment' which takes it to a page to add a description. After you add the description, it creates the appointment, and you have the option to go back to the page to search clients.

If the user desires to see a list of all the consultations and appointments for that particular client, it takes you to a page where the appointments and consultations are listed separately. It is also possible to see detailed information about a particular appointment or consultation by selecting the link "GET INFO" on the respective tables.

Besides, it is also possible to add information to an appointment and convert it into a consultation by selecting the reference "+ADD info". The user starts by adding SOAP notes. Once the SOAP notes are added, the user is directed into a page where multiple nurses (only the ones available at that date and time), diagnostics and prescriptions can

be added. When finished adding information the option "Back into Appointment/Consultation List", returns to to the appointment and consultation list and no more information can be added to that appointment since it is now a consultation.

When seeing all the detailed information about a particular consultation if it has a "dental charting" type of procedure, it is possible to add information regarding a description and measure of a single tooth by selecting the reference "Edit Info". The user is directed into a page where a form containing all teeth and the input fields "description" and "measure" for each tooth is available. If no description or measure is added, the description and measure added into the data base will be "No Description added" and 0.00, respectively. Once all measures are added, the user can go back to appointment and consultation list.

2 Functions, Triggers and Stored Procedures

2.1 1st Task

In this task was asked to write a trigger that computes the age of a client, given is birth date, whenever an appointment is inserted. The code for this trigger is the following.

```
delimiter $$
create trigger update_age after insert on appointment
for each row
begin
    update client
    set client.age = extract(year from (from_days(DATEDIFF(
    current_timestamp,client.birth_date))))
    where client.VAT = new.VAT_client;
end$$
delimiter ;
```

This trigger is only fired after the insert of an appointment, and updates the age of the client by subtracting his birth year to the current year.

2.1.1 Output

As it can be seen in the figure above, the client called Tiago had an original age of 30 years, which did not correspond with his birth year. After inserting an appointment for him, the age updated with real value which was a subtraction between the current year (2019) and his birth year (1965).

2.2 2nd Task

This task was divided into 2 parts: the first, ensuring that a doctor cannot be simultaneously a nurse or a receptionist and vice-versa; the second, ensuring that permanent doctor cannot be simultaneously a trainee doctor and vice versa. The code for both of the them is shown below.

```
1  delimiter $$
2  create trigger check_doctor_nurse_recep_insert before insert on doctor
3  for each row
4  begin
5      if exists (select new.VAT from nurse where new.VAT = nurse.VAT) then
6          SIGNAL SQLSTATE '45000'
7          SET MESSAGE_TEXT = 'Cannot insert doctor if is a nurse';
8      elseif exists (select new.VAT from receptionist where new.VAT =
       receptionist.VAT) then
9          SIGNAL SQLSTATE '45000'
10         SET MESSAGE_TEXT = 'Cannot insert doctor if is a receptionist';
```

```
11        end if;
12   end$$
13   delimiter ;
14
15   delimiter $$
16   create trigger check_doctor_nurse_recep_update before update on doctor
17   for each row
18   begin
19       if exists (select new.VAT from nurse where new.VAT = nurse.VAT) then
20           SIGNAL SQLSTATE '45000'
21           SET MESSAGE_TEXT = 'Cannot update doctor if is a nurse';
22       elseif exists (select new.VAT from receptionist where new.VAT =
         receptionist.VAT) then
23           SIGNAL SQLSTATE '45000'
24           SET MESSAGE_TEXT = 'Cannot update doctor if is a receptionist';
25       end if;
26   end$$
27   delimiter ;
28
29   delimiter $$
30   create trigger check_nurse_doctor_insert before insert on nurse
31   for each row
32   begin
33       if exists (select new.VAT from doctor where new.VAT = doctor.VAT)
         then
34           SIGNAL SQLSTATE '45000'
35           SET MESSAGE_TEXT = 'Cannot insert nurse if is a doctor';
36       end if;
37   end$$
38   delimiter ;
39
40   delimiter $$
41   create trigger check_nurse_doctor_update before update on nurse
42   for each row
43   begin
44       if exists (select new.VAT from doctor where new.VAT = doctor.VAT)
         then
45           SIGNAL SQLSTATE '45000'
46           SET MESSAGE_TEXT = 'Cannot update nurse if is a doctor';
47       end if;
48   end$$
49   delimiter ;
```

```
50
51 delimiter $$
52 create trigger check_receptionist_doctor_insert before insert on
      receptionist
53 for each row
54 begin
55     if exists (select new.VAT from doctor where new.VAT = doctor.VAT)
      then
56         SIGNAL SQLSTATE '45000'
57         SET MESSAGE_TEXT = 'Cannot insert receptionist if is a doctor';
58     end if;
59 end$$
60 delimiter ;
61
62 delimiter $$
63 create trigger check_receptionist_doctor_update before update on
      receptionist
64 for each row
65 begin
66     if exists (select new.VAT from doctor where new.VAT = doctor.VAT)
      then
67         SIGNAL SQLSTATE '45000'
68         SET MESSAGE_TEXT = 'Cannot update receptionist if is a doctor';
69     end if;
70 end$$
71 delimiter ;
```

```
1 delimiter $$
2 create trigger check_permanent_doctor_insert before insert on
      permanent_doctor
3 for each row
4 begin
5     if exists (select new.VAT from trainee_doctor where new.VAT =
      trainee_doctor.VAT) then
6         SIGNAL SQLSTATE '45000'
7         SET MESSAGE_TEXT = 'Cannot insert permamnet doctor if is a
      trainee doctor';
8     end if;
9 end$$
10 delimiter ;
11
12 delimiter $$
13 create trigger check_permanent_doctor_update before update on
```

```
        permanent_doctor
14 for each row
15 begin
16     if exists (select new.VAT from trainee_doctor where new.VAT =
       trainee_doctor.VAT) then
17         SIGNAL SQLSTATE '45000'
18         SET MESSAGE_TEXT = 'Cannot update permamnet doctor if is a
       trainee doctor';
19     end if;
20 end$$
21 delimiter ;
22
23 delimiter $$
24 create trigger check_trainee_doctor_insert before insert on
       trainee_doctor
25 for each row
26 begin
27     if exists (select new.VAT from permanent_doctor where new.VAT =
       permanent_doctor.VAT) then
28         SIGNAL SQLSTATE '45000'
29         SET MESSAGE_TEXT = 'Cannot insert trainee doctor if is a
       permanent doctor';
30     end if;
31 end$$
32 delimiter ;
33
34 delimiter $$
35 create trigger check_trainee_doctor_update before update on
       trainee_doctor
36 for each row
37 begin
38     if exists (select new.VAT from permanent_doctor where new.VAT =
       permanent_doctor.VAT) then
39         SIGNAL SQLSTATE '45000'
40         SET MESSAGE_TEXT = 'Cannot update trainee doctor if is a
       permanent doctor';
41     end if;
42 end$$
43 delimiter ;
```

For the first part, it is needed a trigger before inserting and updating a nurse, receptionist or a doctor, so it was needed 6 triggers to cover all these cases. All six of

the triggers have the same function, which basically checks if the VAT of the employee inserted is already in the tables that are not supposed. If it is, the trigger send a signal of error and prevents the insertion of the employee.

The second part had a similar behaviour with the previous one, which check if the VAT of the doctor inserted, was in the table permanent_doctor or trainee_doctor, and sends an error if it is the case. We also had to make that we had triggers before inserting and updating, for both tables. So in the end were needed 4 triggers.

2.2.1 Output

For the first part, it can be observed in the images below that all cases are checked.

```
MySQL [ist187024]> select * from doctor;
+-----------+---------------+------------------------------------------------+----------------------+
| VAT       | specialization | biography                                      | email                |
+-----------+---------------+------------------------------------------------+----------------------+
|  11111222 | Ears          | Razoável e experiente doutor estudou na nova   | doctor4@gmail.com    |
|  12345876 | Feet          | Ótimo doutor estudou no IST                    | doctor1@gmail.com    |
|  12456876 | Hands         | Péssimo doutor estudou no ISEL                 | doctor2@gmail.com    |
|  14444876 | Stomach       | Good trainee                                   | doctor208@gmail.com  |
|  44440000 | Fingers       | Studied in Harvard                             | doctor200@gmail.com  |
|  55555555 | Nails         | Studied in Portugal FCT                        | doctor209@gmail.com  |
|  88888888 | Teeth         | Excelente doutor estudou no IST                | doctor3@gmail.com    |
| 222000333 | Dead People   | The worst I have ever seen                     | doctor210@gmail.com  |
+-----------+---------------+------------------------------------------------+----------------------+
8 rows in set (0.00 sec)

MySQL [ist187024]> select * from nurse;
+----------+
| VAT      |
+----------+
| 54445867 |
| 54445869 |
| 54445870 |
| 54445875 |
+----------+
4 rows in set (0.00 sec)

MySQL [ist187024]> select * from receptionist;
+----------+
| VAT      |
+----------+
| 14345876 |
| 14445876 |
+----------+
2 rows in set (0.00 sec)

MySQL [ist187024]> Insert into nurse values(88888888);
ERROR 1644 (45000): Cannot insert nurse if is a doctor
MySQL [ist187024]>
MySQL [ist187024]> Insert into receptionist values(88888888);
ERROR 1644 (45000): Cannot insert receptionist if is a doctor
MySQL [ist187024]>
MySQL [ist187024]> update nurse
    -> set nurse.VAT = 88888888
    -> where nurse.VAT = 54445875;
ERROR 1644 (45000): Cannot update nurse if is a doctor
MySQL [ist187024]>
MySQL [ist187024]> update receptionist
    -> set receptionist.VAT = 88888888
    -> where receptionist.VAT = 14445876;
ERROR 1644 (45000): Cannot update receptionist if is a doctor
MySQL [ist187024]>
MySQL [ist187024]>
+-----------+---------------+------------------------------------------------+----------------------+
| VAT       | specialization | biography                                      | email                |
+-----------+---------------+------------------------------------------------+----------------------+
|  11111222 | Ears          | Razoável e experiente doutor estudou na nova   | doctor4@gmail.com    |
|  12345876 | Feet          | Ótimo doutor estudou no IST                    | doctor1@gmail.com    |
|  12456876 | Hands         | Péssimo doutor estudou no ISEL                 | doctor2@gmail.com    |
|  14444876 | Stomach       | Good trainee                                   | doctor208@gmail.com  |
|  44440000 | Fingers       | Studied in Harvard                             | doctor200@gmail.com  |
|  55555555 | Nails         | Studied in Portugal FCT                        | doctor209@gmail.com  |
|  88888888 | Teeth         | Excelente doutor estudou no IST                | doctor3@gmail.com    |
| 222000333 | Dead People   | The worst I have ever seen                     | doctor210@gmail.com  |
+-----------+---------------+------------------------------------------------+----------------------+
8 rows in set (0.00 sec)

MySQL [ist187024]> select * from nurse;
+----------+
| VAT      |
+----------+
| 54445867 |
| 54445869 |
| 54445870 |
| 54445875 |
+----------+
4 rows in set (0.00 sec)

MySQL [ist187024]> select * from receptionist;
+----------+
| VAT      |
+----------+
| 14345876 |
| 14445876 |
+----------+
2 rows in set (0.00 sec)

MySQL [ist187024]> Insert into doctor values(54445867, 'Feet', 'Nurse wants to be a doctor',  'email@gmail.com');
ERROR 1644 (45000): Cannot insert doctor if is a nurse
MySQL [ist187024]>
MySQL [ist187024]> Insert into doctor values(14345876, 'Feet', 'Receptionist wants to be a doctor',  'email@gmail.com');
ERROR 1644 (45000): Cannot insert doctor if is a receptionist
MySQL [ist187024]>
MySQL [ist187024]> update doctor
    -> set doctor.VAT = 54445875
    -> where doctor.VAT = 88888888;
ERROR 1644 (45000): Cannot update doctor if is a nurse
MySQL [ist187024]>
MySQL [ist187024]> update doctor
    -> set doctor.VAT = 14345876
    -> where doctor.VAT = 88888888;
ERROR 1644 (45000): Cannot update doctor if is a receptionist
MySQL [ist187024]>
MySQL [ist187024]>
```

For the second part, all the possible cases are shown.

2.3 3rd Task

In this task was asked to make sure that no individual would have the same phone number. For this, it is needed to check, before inserting and updating, the tables phone_number_employ and phone_number_client, if the the new phone number to be inserted exists in any of those tables. It was 4 triggers that covered all these cases. The code is shown below.

```
delimiter $$
create trigger check_phone_employee_insert before insert on
    phone_number_employee
for each row
begin
    if exists (select phone from phone_number_employee where phone = new
    .phone) then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'An employee already has that phone number';
```

```
 8       elseif exists(select phone from phone_number_client where phone =
     new.phone) then
 9           SIGNAL SQLSTATE '45000'
10           SET MESSAGE_TEXT = 'A client already has that phone number';
11       end if;
12   end$$
13   delimiter ;
14
15   delimiter $$
16   create trigger check_phone_employee_update before update on
     phone_number_employee
17   for each row
18   begin
19       if exists (select phone from phone_number_employee where phone = new
     .phone) then
20           SIGNAL SQLSTATE '45000'
21           SET MESSAGE_TEXT = 'An employee already has that phone number';
22       elseif exists(select phone from phone_number_client where phone =
     new.phone) then
23           SIGNAL SQLSTATE '45000'
24           SET MESSAGE_TEXT = 'A client already has that phone number';
25       end if;
26   end$$
27   delimiter ;
28
29   delimiter $$
30   create trigger check_phone_client_insert before insert on
     phone_number_client
31   for each row
32   begin
33       if exists (select phone from phone_number_client where phone = new.
     phone) then
34           SIGNAL SQLSTATE '45000'
35           SET MESSAGE_TEXT = 'A client already has that phone number';
36       elseif exists(select phone from phone_number_employee where phone =
     new.phone) then
37           SIGNAL SQLSTATE '45000'
38           SET MESSAGE_TEXT = 'An employee already has that phone number';
39       end if;
40   end$$
41   delimiter ;
42
```

```
43 delimiter $$
44 create trigger check_phone_client_update before update on
       phone_number_client
45 for each row
46 begin
47      if exists (select phone from phone_number_client where phone = new.
       phone) then
48          SIGNAL SQLSTATE '45000'
49          SET MESSAGE_TEXT = 'A client already has that phone number';
50      elseif exists(select phone from phone_number_employee where phone =
       new.phone) then
51          SIGNAL SQLSTATE '45000'
52          SET MESSAGE_TEXT = 'An employee already has that phone number';
53      end if;
54 end$$
55 delimiter ;
```

2.3.1 Output

In the following figures all eight possible cases are can observed.

```
MySQL [ist187024]> select * from phone_number_employee;
+-----------+--------------+
| VAT       | phone        |
+-----------+--------------+
|  11111222 | 351922232333 |
|  12345876 | 351911111111 |
|  12456876 | 351911111333 |
|  14345876 | 351914455443 |
|  14444876 | 351914441393 |
|  14445876 | 351914455533 |
|  44440000 | 351914441333 |
|  54445867 | 351964455588 |
|  54445869 | 351964455576 |
|  54445870 | 351964455571 |
|  54445875 | 351964455533 |
|  55555555 | 351914441353 |
|  88888888 | 351911122333 |
| 222000333 | 351916641333 |
+-----------+--------------+
14 rows in set (0.00 sec)

MySQL [ist187024]> select * from phone_number_client;
+-----------+--------------+
| VAT       | phone        |
+-----------+--------------+
|  12345678 | 351917654321 |
|  18273645 | 351922224321 |
|  23456789 | 351917754321 |
|  43215678 | 351922254321 |
|  45678123 | 351966654321 |
|  87333321 | 351937666421 |
|  87654321 | 351937754421 |
|  89123456 | 351937754321 |
+-----------+--------------+
8 rows in set (0.00 sec)

MySQL [ist187024]> Insert into phone_number_employee values(88888888,  351911111111);
ERROR 1644 (45000): An employee already has that phone number
MySQL [ist187024]>
MySQL [ist187024]> Insert into phone_number_employee values(88888888, 351917654321);
ERROR 1644 (45000): A client already has that phone number
MySQL [ist187024]>
MySQL [ist187024]> update phone_number_employee
    -> set phone_number_employee.phone = 351911111111
    -> where phone_number_employee.VAT = 88888888;
ERROR 1644 (45000): An employee already has that phone number
MySQL [ist187024]>
MySQL [ist187024]> update phone_number_employee
    -> set phone_number_employee.phone = 351917654321
    -> where phone_number_employee.VAT = 88888888;
ERROR 1644 (45000): A client already has that phone number
MySQL [ist187024]>
```

2.4   4th Task

In this task, the objective is to make a funtion in SQL that returned the total number of appointments that did not result in a consultation, given some parameters such as the year of the appointment, the gender of the client and the age limits (upper and lower limits). The function made can be observed below.

```
1  DELIMITER $$
2  create function no_shows(gender varchar(20), year_ int, upper_age int,
      lower_age int)
3  returns int
4
5  begin
6
```

```
 7 declare no_shows int;
 8
 9 select count(*) into no_shows from (
10 select a.VAT_doctor, a.date_timestamp, a.VAT_client from appointment as
       a, client as cl
11 where not exists (select VAT_doctor, date_timestamp
12 from consultation as c where c.VAT_doctor = a.VAT_doctor and c.
       date_timestamp = a.date_timestamp)
13 and extract(year from a.date_timestamp) = year_
14 and a.VAT_client = cl.VAT
15 and cl.gender = gender
16 and cl.age > lower_age and cl.age < upper_age
17 group by a.VAT_doctor, a.date_timestamp) as x;
18
19 return no_shows;
20
21 end$$
22 DELIMITER ;
```

This function has as four input arguments, which are year, gender, upper and lower limits for the age of the client. In the function, it is executed a query that counts the total number of appointments that did not result in a consultation, taking into consideration the input parameters. At the end, it returns that total.

2.4.1 Output

In the picture below, it can be observed all the appointments that did not result in consultation, together with the client age and gender.

It can be observed that, for example in the year 2019 there were 9 appointments that did not turn into consultations where the client is a female and has 14 years old. On

the other hand, in the following year only one appointment did not convert, for a male with 54 years old.

Executing the function, it can be concluded that it works as expected. It can be checked below.

2.5   5th Task

In this task, it was asked to write a stored procedure to change the salary of doctors that have been working for a determined number of years. In case he had given 100 or more consultations in the current year, the salary increase would be 10%, otherwise 5%. The code for the stored procedure can be observed below.

```
1  DELIMITER $$
2  CREATE PROCEDURE raise_doctor_salary(in years int)
3
4  begin
5      declare num_cons int;
6      declare done int default 0;
7      declare r int;
8      declare c_salary numeric(20,2);
9
10     declare c cursor FOR
11         select e.VAT from employee as e, doctor as d, permanent_doctor
       as pd where pd.VAT = d.VAT and d.VAT = e.VAT and pd.years > years;
12
13     declare continue handler
14         for not found set done = 1;
15
```

```
16      open c;
17      repeat
18          fetch c into r;
19          if not done then
20              select count(*) into num_cons from
21              (select VAT_doctor, date_timestamp from consultation where
    VAT_doctor = r
22              and extract(year from date_timestamp) = extract(year from
    current_timestamp)) as count_cons;
23
24              select salary into c_salary from employee where VAT = r;
25
26              if num_cons >= 100 then
27                  set c_salary = c_salary*1.1;
28              else
29                  set c_salary = c_salary*1.05;
30              end if;
31
32              update employee
33              set employee.salary = c_salary
34              where employee.VAT = r;
35
36          end if;
37          until done
38          end repeat;
39          close c;
40 end$$
41 DELIMITER ;
```

The stored procedure takes in as an input parameter an integer which is the number of years and executes a creates a cursor for a query that gives the VAT of the doctors that have been working for more than the number of years in the input parameter. After that it iterates through the VAT in that query (by means of the cursor) and executes a query to check if the doctor has more than 100 consultations. If yes, a variable with the salary of the doctor is set to 10% higher, if not, only 5%, then it updates the salary of the current doctor.

2.5.1 Output

In the query below, it can be observed the doctors with their respective original salary and the number of years they have been working.

```
MySQL [ist187024]> select e.name_, e.VAT, e.salary, pd.years  from employee as e, doctor as d, permanent_doctor as pd  where e.VAT = d.VAT and d.VAT = pd.VAT;
+----------------+----------+---------+-------+
| name_          | VAT      | salary  | years |
+----------------+----------+---------+-------+
| Jack Sweettath | 11111222 | 3000.00 |    15 |
| Joao           | 12345876 | 1300.00 |    10 |
| Rui            | 12456876 | 1300.00 |     6 |
| Jane Sweettooth| 88888888 | 2500.00 |     8 |
+----------------+----------+---------+-------+
4 rows in set (0.00 sec)
```

Let's test for the stored procedure for doctors that have been working for 7 or more years. In this case there are three doctors that meet that requirement. Now it is needed to be checked the number of consultations for each doctor in the current year, which can be observed in the picture below.

```
MySQL [ist187024]> select count(*) as num_consultations from (select VAT_doctor, date_timestamp from consultation
    ->                                         where VAT_doctor = 12345876
    ->                                         and extract(year from date_timestamp) = 2019
    ->                                         group by VAT_doctor, date_timestamp) as x;
+-------------------+
| num_consultations |
+-------------------+
|                 2 |
+-------------------+
1 row in set (0.00 sec)

MySQL [ist187024]> select count(*) as num_consultations from (select VAT_doctor, date_timestamp from consultation
    ->                                         where VAT_doctor = 11111222
    ->                                         and extract(year from date_timestamp) = 2019
    ->                                         group by VAT_doctor, date_timestamp) as x;
+-------------------+
| num_consultations |
+-------------------+
|                 0 |
+-------------------+
1 row in set (0.00 sec)

MySQL [ist187024]> select count(*) as num_consultations from (select VAT_doctor, date_timestamp from consultation
    ->                                         where VAT_doctor = 88888888
    ->                                         and extract(year from date_timestamp) = 2019
    ->                                         group by VAT_doctor, date_timestamp) as x;
+-------------------+
| num_consultations |
+-------------------+
|               102 |
+-------------------+
1 row in set (0.00 sec)
```

As it can observed, only one doctor (with the VAT 88888888, correspondent to Jane Sweettooth) has 100 or more consultations. In the end only this doctor should be raised 10%, and the other two 5%.

In the following figure, it can be observed the result of the stored procedure, which raised the salaries as expected.

```
MySQL [ist187024]> call raise_doctor_salary(7);
Query OK, 0 rows affected, 1 warning (0.01 sec)

MySQL [ist187024]>  select e.name_, e.VAT, e.salary, pd.years  from employee as e, doctor as d, permanent_doctor as pd  where e.VAT = d.VAT and d.VAT = pd.VAT;
+----------------+----------+---------+-------+
| name_          | VAT      | salary  | years |
+----------------+----------+---------+-------+
| Jack Sweettath | 11111222 | 3150.00 |    15 |
| Joao           | 12345876 | 1365.00 |    10 |
| Rui            | 12456876 | 1300.00 |     6 |
| Jane Sweettooth| 88888888 | 2750.00 |     8 |
+----------------+----------+---------+-------+
4 rows in set (0.00 sec)
```