



SISTEMAS DE INFORMAÇÃO E BASE DE DADOS

Relatório da Parte 3 do Projecto

Autores: Grupo 14

Sophie Taboada 84187

Helena Oliveira 87017

Tiago Cabecinha 87124

Professores:

Bruno Emanuel Da Graça Martins

João Granado Marques

6 de Dezembro de 2019

Conteúdo

1	Introdução e Objectivos	3
2	Arquitectura da Aplicação móvel	4
3	Aplicação Web Usando a Base de dados	5
3.1	Procurar e Apresentação dos Clientes	5
3.1.1	Mecanismos de Procura - search_client.php	5
3.1.2	Apresentação dos Dados dos Clientes - search_client.php	6
3.2	Adicionar Clientes	7
3.3	Marcações e Consultas de um cliente	8
3.3.1	Adicionar Marcação	8
3.3.2	Detalhes das Marcações e Consultas	10
3.3.3	Adicionar detalhes a Marcações	12
3.3.4	Adicionar Dental Charting	14
4	Funções, Triggers e <i>Procedures</i> guardadas	15
4.1	Task 1	16
4.2	Task 2	17
4.2.1	a)	17
4.2.2	b)	20
4.3	Task 3	21
4.4	Task 4	24
4.5	Task 5	24
5	Conclusão	26
6	Anexos	27
6.1	Inicial Page	27
6.1.1	inicial_page.php	27
6.2	Add clients	28
6.2.1	new_client_form.php	28
6.2.2	and new_client.php	29
6.3	Search for and display Clients	32
6.3.1	search_client_form.php	32
6.3.2	search_client.php	33
6.4	Appointments and Consultations	38
6.4.1	apps_and_cons.php	38
6.4.2	list_cons_info.php	41
6.5	Book Appointment	49
6.5.1	add_app_form.php	49
6.5.2	book_app.php	50
6.5.3	search_doc.php	52
6.6	Add information to an Appointment	55
6.6.1	add_info_form.php	55
6.6.2	add_soap.php	65
6.6.3	add_nurse.php	66
6.6.4	add_diagnostic.php	68

6.6.5	add_prescription.php	69
6.7	Insert Dental Charting	72
6.7.1	add_dental_form.php	72
6.7.2	add_dental.php	77

1 Introdução e Objectivos

Nesta terceira parte do projeto é pedido para que, a partir da base de dados criada e populada na segunda parte do projeto, se crie uma Aplicação Web em que se possa realizar um determinado conjunto de acções sobre a base de dados então criada de uma clínica dentária usando as linguagens PHP e HTML. Este trabalho pode ser acedido através do URL: http://web.tecnico.ulisboa.pt/ist425487/initial_page.php. Este URL leva a uma página inicial, um ponto de partida para o utilizador. Para se criar e popular a base de dados, será necessário correr os scripts *database.sql* e *populate.sql*. Os scripts relacionados com a segunda parte deste terceiro projeto são o *triggers.sql* e *functions.sql*. Todas os pontos do enunciado deste terceiro projeto foram desenvolvidos e encontram-se detalhadamente explicados neste relatório.

2 Arquitectura da Aplicação móvel

Na figura 1 pode-se observar a arquitetura do website composto para a terceira parte do projeto. As setas a vermelho indicam o que acontece quando uma certa tecla é primida.

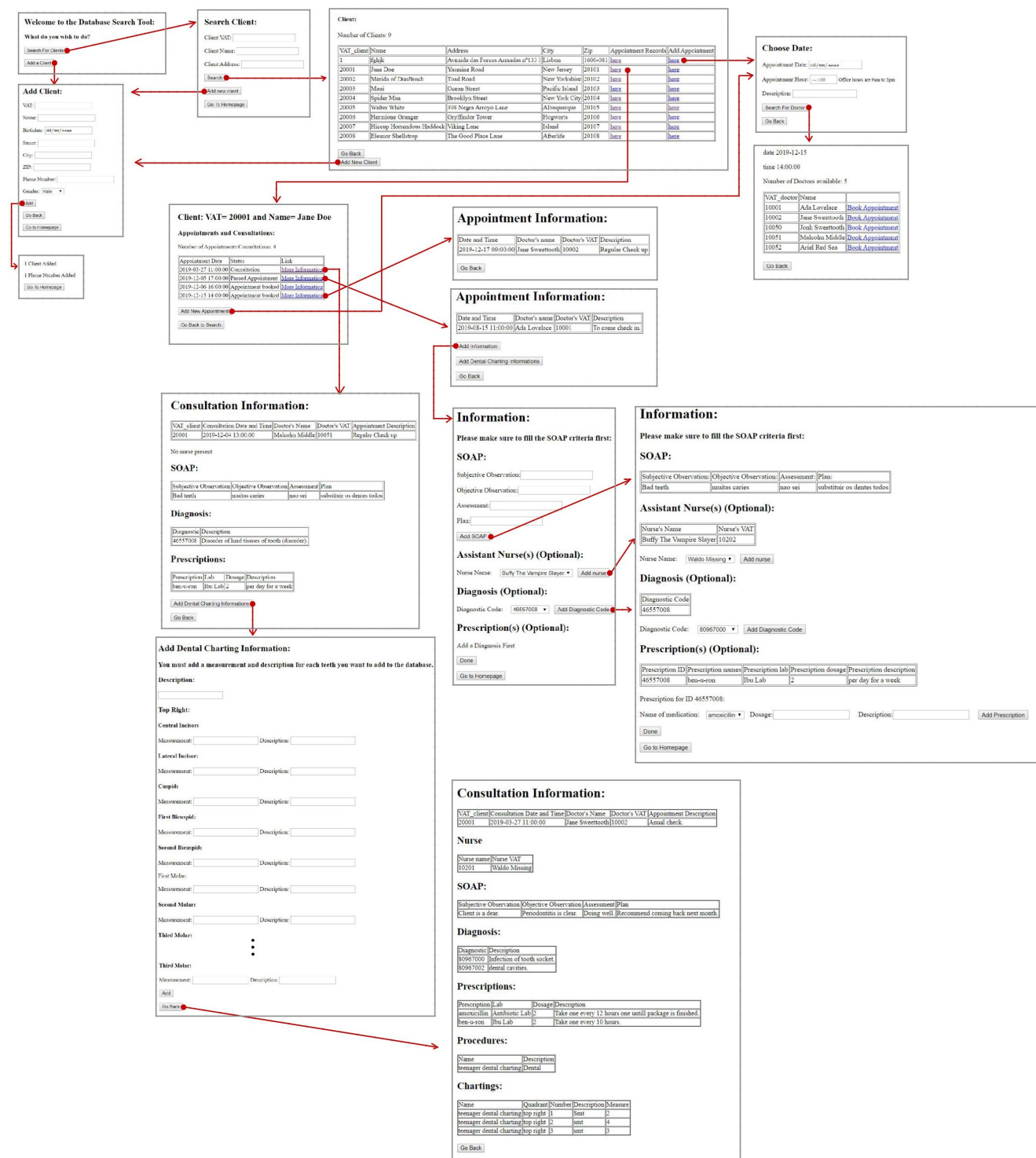


Figura 1: Arquitectura do site.

3 Aplicação Web Usando a Base de dados

3.1 Procurar e Apresentação dos Clientes

A página WEB inicial da aplicação consiste numa página muito simples com dois botões como se pode observar na figura 2.

Welcome to the Database Search Tool:

What do you wish to do?

Search For Clients

Add a Client

Figura 2: Página Inicial

O primeiro, como o nome indica direcciona o utilizador para uma pagina de pesquisa onde este pode procurar a informação de um cliente a partir do seu VAT completo, ou através da totalidade ou parte do seu nome e/ou morada. Para isso são efectuados diferentes *queries* segundo os campos que se encontram preenchidos.

O segundo botão permite ao utilizador adicionar um novo cliente à base de dados.

3.1.1 Mecanismos de Procura - search_client.php

A página que é apresentada quando se deseja efectuar a procura de um cliente é apresentada na figura 3. Existem 3 campos que podem ser preenchidos cuja informação utilizada para fazer a filtragem dos clientes.

Search Client:

Client VAT:

Client Name:

Client Address:

Search

Add new client

Go To Homepage

Figura 3: Procura de Cliente

Para o caso em que apenas o VAT do cliente se encontra preenchido, é possível detectar directamente se o cliente existe comparando o VAT com a base de dados. Para o caso em

que o nome ou a morada são parcialmente ou inteiramente preenchidos é efectuado um "LIKE" para a filtragem em SQL para identificar se estes campos coincidem com algum da base de dados. Abaixo no *listing 1* encontra-se o exemplo do query em que todos os campos foram preenchidos.

Listing 1: Query de Procura de um Cliente

```

1 SELECT *
2 FROM client
3
4 /*Se o campo VAT_client estiver preenchido*/
5 WHERE VAT_client =:VAT\_client
6
7 /*Se o campo Client name estiver preenchido*/
8 AND name LIKE CONCAT('%',:client\_name,'%')
9
10 /*Se o campo Street name estiver preenchido*/
11 AND street LIKE CONCAT('%',:client\_addr,'%') OR zip LIKE CONCAT('%',:
    client\_addr,'%') OR city LIKE CONCAT('%',:client\_addr,'%')

```

Nos casos em nem todos os campos são preenchidos, são apenas usados *queries* cujos filtros correspondem aos dos campos preenchidos. Finalmente, para o caso em que nenhum dos campos é preenchido, é apresentada a informação acerca de todos os clientes.

3.1.2 Apresentação dos Dados dos Clientes - search_client.php

A lista dos clientes é apresentada no formato da figura 4, indicando primeiro o VAT, depois o nome, e por seguinte a morada, cidade e Zip. Esta informação é retirada através do query do *listing 1* que identifica o(s) cliente(s) correspondentes à busca e retorna toda a informação acerca deste(s).

Client:

Number of Clients: 9

VAT_client	Name	Address	City	Zip	Appointment Records	Add Appointment
1	fghjk	Avenida das Forcas Armadas n°133 l	Lisbon	1600-081	here	here
20001	Jane Doe	Yasmine Road	New Jersey	20101	here	here
20002	Merida of DunBroch	Toad Road	New Yorkshire	20102	here	here
20003	Maui	Ocean Street	Pacific Island	20103	here	here
20004	Spider Man	Brooklyn Street	New York City	20104	here	here
20005	Walter White	308 Negra Arroyo Lane	Albuquerque	20105	here	here
20006	Hermione Granger	Gryffindor Tower	Hogwarts	20106	here	here
20007	Hiccup Horrendous Haddock	Viking Lane	Island	20107	here	here
20008	Eleanor Shellstrop	The Good Place Lane	Afterlife	20108	here	here

[Go Back](#)

[Add New Client](#)

Figura 4: Lista dos clientes após a procura

Nesta página pode se escolher ver informação sobre as marcações ou consultas de cada cliente: "Appointment Records", ou então fazer uma marcação: "Add Appointment". É

também possível voltar para atrás de volta para a pesquisa ou então adicionar um novo cliente.

3.2 Adicionar Clientes

O botão adicionar clientes redirecciona o utilizador para um formulário como o da figura 5.

Add Client:

VAT:

Name:

Birthdate:

Street:

City:

ZIP:

Phone Number:

Gender: ▼

Figura 5: Formulário de adicionar clientes

Todos os campos são obrigatórios, as datas de nascimento permitidas são desde o ano 1910 até ao dia actual. Caso o VAT já tenha sido utilizado, o utilizador é informado e o cliente não é criado. Caso o utilizador deseje apenas adicionar um número de telefone diferente para um dado cliente, este pode preencher o formulário com os dados do cliente e o seu novo numero. Este será registado na base de dados. Quando o utilizador carrega no botão "Add" e tudo se apresenta de acordo com o correcto, é apresentada uma página de confirmação da criação, figura 6.

1 Client Added

1 Phone Number Added

Figura 6: Confirmação de cliente adicionado na Base de Dados

Após o cliente ser criado existe um botão para voltar para a página inicial.

3.3 Marcações e Consultas de um cliente

Se o utilizador optar por ver as marcações e consultas de um determinado cliente, este carrega no hyperlink "here" da página *search_client.php* apresentada na figura 4 e a página é redireccionada para a página *apps_cons_info.php* cuja apresentação figura na figura 7.

Client: VAT= 20001 and Name= Jane Doe

Appointments and Consultations:

Number of Appointments/Consultations: 4

Appointment Date	Status	Link
2019-03-27 11:00:00	Consultation	More Information
2019-12-04 11:00:00	Consultation	More Information
2019-12-04 13:00:00	Passed Appointment	More Information
2019-12-11 10:00:00	Appointment booked	More Information

Add New Appointment

Go Back to Search

Figura 7: Apresentação das consultas e marcações do cliente

As marcações cuja data é inferior à do dia actual são marcadas como "Passed Appointments" se ainda não tiverem sido adicionadas informações acerca da consulta. As marcações cuja data é superior à data actual são indicadas como "Booked Appointments". É possível adicionar informações sobre *SOAP*, assistentes, diagnósticos e prescrições nas marcações "Passed Appointments" mas não nas "Booked Appointments" pois não faz sentido adicionar informação a uma marcação cuja consulta ainda não ocorreu. Uma vez que é adicionada informação a uma "Passed Appointment" (obrigatoriamente os "SOAP"), a marcação passa a ser considerada como consulta. Não é possível modificar ou eliminar a informação das consultas à excepção do *dental charting*.

3.3.1 Adicionar Marcação

Carregando no Hyperlink "Add" na página 4 ou no botão "Add New Appointment" da página da figura 7, o utilizador é redireccionado para uma página com um formulário correspondendo ao da figura 8

Choose Date:

Appointment Date:

Appointment Hour: Office hours are 9am to 5pm

Description:

Figura 8: Formulário para adicionar consulta

As marcações só podem ser feitas a partir do dia actual da marcação e as horas da marcação estão restringidas das 9h00 da manhã as 17h00, sendo o intervalo de cada marcação uma hora. Para efectuar isso, utilizou-se os requisitos no html apresentados no *listing 2*.

Listing 2: Peça do html utilizado para o form de adicionar marcações

```
1 ...
2 <p>Appointment Date:
3   <input type="date" name="app_date" min='<?php echo date("Y-m-d");?>'
   required/>
4 </p>
5 <p>Appointment Hour:
6   <input type="time" name="app_time" min="09:00:00" max="17:00:00"
   step="3600" required/>
7   <small>Office hours are 9am to 5pm</small>
8 </p>
9 ...
```

Quando os campos estão preenchidos é necessário procurar a disponibilidade dos médicos para esse dia e horário, o utilizador é redireccionado para a página apresentada na figura 9. Nesta página constam os médicos que estão disponíveis para os horários indicados para a consulta.

date 2019-12-15

time 14:00:00

Number of Doctors available: 5

VAT_doctor	Name	
10001	Ada Lovelace	Book Appointment
10002	Jane Sweettooth	Book Appointment
10050	Jonh Sweettooth	Book Appointment
10051	Malcolm Middle	Book Appointment
10052	Ariel Red Sea	Book Appointment

Go Back

Figura 9: Médicos Disponíveis

Quando o utilizador carrega no botão "book Appointment", a Marcação é inserida na base de dados através do query do listing 3

Listing 3: Peçaço do html utilizado para adicionar consultas

```

1 INSERT INTO appointment (VAT_doctor, date_timestamp, description,
   VAT_client)
2 VALUES (:VAT_doctor, :datetime_new, :description, :VAT_client)

```

3.3.2 Detalhes das Marcações e Consultas

É possível observar os detalhes referentes às consultas e marcações carregando no hyper-link "More Information" da página vista na figura 7. O display é diferente para os três casos: "Consultation", "Passed Appointment" e "Appontment Booked".

Para o caso da consulta é apresentada uma página com todas as informações disponíveis e, se ainda não existir, é permitido ao utilizador adicionar um *dental charting*.

Consultation Information:

VAT_client	Consultation Date and Time	Doctor's Name	Doctor's VAT	Appointment Description
20002	2018-02-27 11:00:00	Ada Lovelace	10001	To eliminate rotti

Nurse

Nurse name	Nurse VAT
10201	Waldo Missing

SOAP:

Subjective Observation	Objective Observation	Assessment	Plan
Client is redhead.	Has nothing else wrong.	Doing well	Recommend no sugar.

Diagnosis:

Diagnostic	Description
80967002	dental cavities.

Prescriptions:

Prescription	Lab	Dosage	Description
ibuprofen	Ibu Lab	2	Take one every 8 hours.

Add Dental Charting Informations

Go Back

(a) Sem Dental Charting

Consultation Information:

VAT_client	Consultation Date and Time	Doctor's Name	Doctor's VAT	Appointment Description
20001	2019-03-27 11:00:00	Jane Sweettooth	10002	Annual check.

Nurse

Nurse name	Nurse VAT
10201	Waldo Missing

SOAP:

Subjective Observation	Objective Observation	Assessment	Plan
Client is a dear.	Periodontitis is clear.	Doing well.	Recommend coming back next month.

Diagnosis:

Diagnostic	Description
80967000	Infection of tooth socket.
80967002	dental cavities.

Prescriptions:

Prescription	Lab	Dosage	Description
amoxicillin	Antibiotic Lab	2	Take one every 12 hours one until package is finished.
ben-u-ron	Ibu Lab	2	Take one every 10 hours.

Procedures:

Name	Description
teenager dental charting	Dental

Chartings:

Name	Quadrant	Number	Description	Measure
teenager dental charting	top right	1	Smt	2
teenager dental charting	top right	2	smt	4
teenager dental charting	top right	3	smt	3

Go Back

(b) Com Dental Charting

Figura 10: Informação da consulta

Para o caso das "Booked Appointments" é possível observar apenas o VAT do cliente, a data da marcação, o nome do médico responsável e o seu VAT e a Descrição da marcação. Não é permitido ao utilizador adicionar informações sobre a consulta ou sobre o *dental charting*, como se pode ver na figura 11.

Appointment Information:

Date and Time	Doctor's name	Doctor's VAT	Description
2019-12-17 09:00:00	Jane Sweettooth	10002	Regular Check up

Go Back

Figura 11: Informação da Marcação ainda não ocorrida

Para o caso das "Passed Appointments" é possível observar apenas o VAT do cliente, a data da marcação, o nome do médico responsável e o seu VAT e a Descrição da marcação.

É permitido ao utilizador adicionar informações sobre a consulta e sobre o *dental charting*, como se pode ver na figura 12.

Appointment Information:

Date and Time	Doctor's name	Doctor's VAT	Description
2019-08-15 11:00:00	Ada Lovelace	10001	To come check in.

Add Information

Add Dental Charting Informations

Go Back

Figura 12: Informação da Marcação já ocorrida

3.3.3 Adicionar detalhes a Marcações

É possível adicionar informação a uma "Passed Appointment" carregando no botão "Add Information" na página da figura 12. O utilizador é redireccionado para uma pagina com o formulário da figura 13:

Information:

Please make sure to fill the SOAP criteria first:

SOAP:

Subjective Observation:

Objective Observation:

Assessment:

Plan:

Assistant Nurse(s) (Optional):

Nurse Name:

Diagnosis (Optional):

Diagnostic Code:

Prescription(s) (Optional):

Add a Diagnosis First

Figura 13: Form para adicionar informação a uma Marcação já ocorrida

Não é possível adicionar enfermeiras, diagnósticos ou prescrições se não for submetido um SOAP antes. Isto permite adicionar o registo da consulta na base de dados para poder adicionar as seguintes informações a seguir. Caso se tente adicionar informações sem as SOAP submetidas, a mensagem de erro da imagem 14 é mostrada.

There isn't a consultation associated to this appointment.

Please create one first by adding SOAP information to the appointment.

Figura 14: Aviso não há consulta ainda

Uma vez as SOAP submetidas pode-se adicionar tantas enfermeiras e diagnósticos quanto desejados. A adição da informação é feita "instantaneamente" após cada submit, i.e., o form é actualizado e apresenta a informação já submetida, não permitindo submeter informação repetida. É apenas permitido inserir prescrições se for submetido pelo menos um diagnostico. A adição de prescrições é feita por diagnóstico como é possível observar na figura 15.

Information:

Please make sure to fill the SOAP criteria first:

SOAP:

Subjective Observation:	Objective Observation:	Assessment:	Plan:
Bad teeth	muitas caries	nao sei	substituir os dentes todos

Assistant Nurse(s) (Optional):

Nurse's Name	Nurse's VAT
Buffy The Vampire Slayer	10202

Nurse Name:

Diagnosis (Optional):

Diagnostic Code
46557008

Diagnostic Code:

Prescription(s) (Optional):

Prescription ID	Prescription names	Prescription lab	Prescription dosage	Prescription description
46557008	ben-u-ron	Ibu Lab	2	per day for a week

Prescription for ID 46557008:

Name of medication: Dosage: Description:

Figura 15: Form preenchido para adicionar informação a uma Marcação já ocorrida

3.3.4 Adicionar Dental Charting

Apenas é possível adicionar informações de mapeamento dentário após já se ter inserido as restantes informações base da consulta em questão (as notas SOAP). Caso a consulta ainda não exista na base de dados e o utilizador tente inserir informações dentárias, é dado um aviso, mostrado na figura 14.

Ao carregar no botão *Add Dental Charting Informations*, mostrado na figura 10, o site é reencaminhado para uma página que permite adicionar informação do procedimento. É obrigatório inserir uma descrição geral deste procedimento e depois é permitido inserir medidas e descrições sobre todos os dentes possíveis, mas apenas é necessário inserir informações sobre os dentes desejados catalogar. Esta página encontra-se na figura 16.

Add Dental Charting Information:

You must add a measurement and description for each teeth you want to add to the database.

Description:

Top Right:

Central Incisor:

Measurement: Description:

Lateral Incisor:

Measurement: Description:

Cuspid:

Measurement: Description:

First Bicuspид:

Measurement: Description:

Second Bicuspид:

Measurement: Description:

First Molar:

Measurement: Description:

Second Molar:

Measurement: Description:

Third Molar:

Figura 16: Adicionar informação Dental Charting

Como se observa na figura 16, apenas serão adicionados à base de dados os dados dos dentes em que ambos os campos medida e descrição são preenchidos.

Foi entendido por transação única que apenas havia uma oportunidade para inserir informação de mapeamento de dentes na base de dados. Esta página permite portanto que sejam preenchidos todos os campos dos dentes que o utilizador deseja inserir, e após carregar no botão *Add*, não é permitido adicionar mais dados à informação já submetida, ou seja, deixa de haver a opção *Add Dental Charting Informations*, para esta consulta e não é possível adicionar informação de mais dentes.

4 Funções, Triggers e *Procedures* guardadas

As Funções, os Triggers e as *Stored Procedures* são funcionalidades que se podem aproveitar quando se utiliza uma base de dados. Todas estas funcionalidades são códigos desenvolvidos em SQL que performem uma operação à base de dados e são utilizados para facilitar o uso e o acesso a esta mesmo. As funções são utilizadas para retornar um valor que se possa querer de uma certa tabela da base de dados. As *Stored Procedures* são códigos preparados e guardados para serem usados as vezes que quisermos. Por exemplo, se uma query for executada muitas vezes, pode-se apenas criar uma procedure com essa query e posteriormente chamar essa procedure de modo a simplificar as tarefas. Já os Triggers são também códigos preparados para serem executados, mas como o nome indica, são apenas executados quando é detectada alguma modificação na base de dados.

4.1 Task 1

É pedido que seja criado um Trigger que realize uma actualização da idade de um cliente de acordo com a sua data de nascimento comparando com a data actual sempre que seja adicionada uma *Appointment* na base de dados para este cliente. O código para a execução da Task 1 encontra-se no *listing listing 4*.

Listing 4: Task 1

```
1 CREATE TRIGGER client_age_upd AFTER INSERT ON appointment
2 FOR EACH ROW
3 BEGIN
4     DECLARE most_recent TIMESTAMP;
5     SET most_recent = YEAR(CURDATE());
6
7     UPDATE client
8     SET age = TIMESTAMPDIFF(YEAR, birth_date, most_recent)
9     WHERE VAT_client = NEW.VAT_client;
10 END$$
```

A idade de um cliente é logo atualizada após uma inserção na tabela *Appointment* como é indicado com *AFTER INSERT ON*. De seguida, declara-se uma variável do tipo *TIMESTAMP* e coloca-se nessa mesma variável o ano corrente, aquando da inserção da *Appointment*. O código seguinte, trata de realizar a atualização da idade, encontrando o cliente pelo seu VAT e de seguida, atualiza-se a idade desse cliente realizando a diferença entre o ano corrente e o ano de nascimento.

Para testar a primeira task, foram desenvolvidas umas queries apresentadas no *listing 5*.

Listing 5: Test Task1

```
1 INSERT INTO client
2 VALUES (20019, 'Someone', '19980303', 'Cnx', 'Ptpt', 1004, 'male',
3         10);
4 SELECT age AS old_age FROM client
5 WHERE name = 'Someone' AND VAT_client = 20019;
6 INSERT INTO appointment
7 VALUES (10002, '2020-01-01 10:10:10', 'Meh', 20019);
8 SELECT age AS new_age FROM client
9 WHERE name = 'Someone' AND VAT_client = 20019;
```

```

MySQL [ist187124]> INSERT INTO client
-> VALUES (20019, 'Someone', '19980303', 'Cnx', 'Cnxx', 1004, 'male', 10);
Query OK, 1 row affected (0.00 sec)

MySQL [ist187124]>
MySQL [ist187124]> SELECT age AS old_age FROM client
-> WHERE name = 'Someone' AND VAT_client = 20019;
+-----+
| old_age |
+-----+
|      10 |
+-----+
1 row in set (0.00 sec)

MySQL [ist187124]>
MySQL [ist187124]> INSERT INTO appointment
-> VALUES (10002, '2020-01-01 10:10:10', 'Meh', 20019);
Query OK, 1 row affected (0.01 sec)

MySQL [ist187124]>
MySQL [ist187124]> SELECT age AS new_age FROM client
-> WHERE name = 'Someone' AND VAT_client = 20019;
+-----+
| new_age |
+-----+
|      21 |
+-----+
1 row in set (0.00 sec)

```

Figura 17: Testing task 1

Verifica-se na imagem 17 que a idade do cliente são 10 anos aquando da criação deste mesmo cliente. Após a inserção de uma appointment para este cliente, o trigger da task 1 é disparado e a idade do cliente é atualizada para a idade atual. Verifica-se que a idade foi atualizada de 10 para 21 anos uma vez que a data de nascimento do cliente na qual se marcou uma appointment é de 1998. O mesmo aconteceria com uma atualização de uma appointment para este cliente.

4.2 Task 2

Para a segunda Task é pedido que sejam criados Triggers que assegurem:

4.2.1 a)

Um funcionário que seja ou recepcionista ou uma assistente não poderá ser simultaneamente um doutor. Assumiu-se também o contrário, que um doutor não poderá também ser um recepcionista ou um assistente. Para garantir estas condições, foram criadas várias procedures que fazem uma verificação sempre antes de uma inserção ou uma atualização quer de um doutor, de um recepcionista e de um assistente. Estas procedures são depois chamadas nos respetivos Triggers. O código realizado para a execução da Task 2a) encontra-se no *listing* 6.

Listing 6: Task 2

```

1 CREATE PROCEDURE check_doct
2 (IN VAT INTEGER)
3 BEGIN

```

```

4 DECLARE msg VARCHAR(255);
5 IF EXISTS (SELECT recepcionist.VAT FROM recepcionist
6           WHERE recepcionist.VAT = VAT) THEN
7     SET msg = CONCAT("IC: person VAT ", VAT,
8     " is already a recepcionist, cannot be a doctor too.");
9
10    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
11  END IF;
12  IF EXISTS (SELECT nurse.VAT FROM nurse
13            WHERE nurse.VAT = VAT) THEN
14    SET msg = CONCAT("IC: person VAT ", VAT,
15    " is already a nurse, cannot be a doctor too.");
16
17    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
18  END IF;
19 END$$
20 /*triggers to call procedure to check alterations in doctors table*/
21 CREATE TRIGGER ins_doctor BEFORE INSERT ON doctor
22 FOR EACH ROW
23 BEGIN
24   CALL check_doct(NEW.VAT);
25 END$$
26 CREATE TRIGGER upd_doctor BEFORE UPDATE ON doctor
27 FOR EACH ROW
28 BEGIN
29   CALL check_doct(NEW.VAT);
30 END$$
31 /*procedure to check if a recepcionist is not in doctor table*/
32 CREATE PROCEDURE check_recip
33 (IN VAT INTEGER)
34 BEGIN
35   DECLARE msg VARCHAR(255);
36   IF EXISTS (SELECT doctor.VAT FROM doctor
37             WHERE doctor.VAT = VAT) THEN
38     SET msg = CONCAT("IC: person VAT ", VAT,
39     " is already a doctor, cannot be a recepcionist too.");
40
41     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
42   END IF;
43 END$$
44 /*procedure to check if a nurse is not in doctor table*/
45 CREATE PROCEDURE check_nur
46 (IN VAT INTEGER)
47 BEGIN
48   DECLARE msg VARCHAR(255);
49   IF EXISTS (SELECT doctor.VAT FROM doctor
50             WHERE doctor.VAT = VAT) THEN
51     SET msg = CONCAT("IC: person VAT ", VAT,
52     " is already a doctor, cannot be a nurse too.");
53
54     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
55   END IF;
56 END$$
57 /*triggers to call procedure to check alterations in recepcionist table
58 */
59 CREATE TRIGGER ins_recip BEFORE INSERT ON recepcionist
60 FOR EACH ROW
61 BEGIN

```

```

61     CALL check_recp(NEW.VAT);
62 END$$
63 CREATE TRIGGER upd_recp BEFORE UPDATE ON recepcionist
64 FOR EACH ROW
65 BEGIN
66     CALL check_recp(NEW.VAT);
67 END$$
68 /*triggers to call procedure to check alterations in nurses table*/
69 CREATE TRIGGER ins_nur BEFORE INSERT ON nurse
70 FOR EACH ROW
71 BEGIN
72     CALL check_nur(NEW.VAT);
73 END$$
74 CREATE TRIGGER upd_nur BEFORE UPDATE ON nurse
75 FOR EACH ROW
76 BEGIN
77     CALL check_nur(NEW.VAT);
78 END$$

```

As três procedures têm como argumento o VAT inserido na tentativa inserção ou atualização na respetiva tabela. Para cada situação, são chamadas sempre antes de existir uma inserção (*BEFORE INSERT*) ou uma atualização *BEFORE UPDATE* na tabela a que a procedure está associada. Em cada procedure é declarado uma variável *msg* que indicará uma mensagem de erro no caso de se estar a tentar fazer uma alteração que vá em contra o pedido na Task. A query é realizada através de uma condição *IF* que verifica se existe (*EXISTS*) o VAT de um funcionário numa tabela em que não deveria estar.

Para testar a segunda task, foram desenvolvidas umas queries apresentadas no *listing* 7.

Listing 7: Task2

```

1 /* 2. Test trigger doctors and nurses */
2 INSERT INTO doctor /*insert into doctor a nurse*/
3 VALUES (10202, 'oral medicine', 'Graduated with honors.', 'sweet.
    jane@gmail.com');
4 INSERT INTO doctor /*insert into doctor a recepcionist*/
5 VALUES (10401, 'oral medicine', 'Graduated with honors.', 'sweet.
    jane@gmail.com');

```

Tentou-se inserir na tabela dos doutores 2 pessoas, uma assistente e uma recepcionista em que o trigger detetou que se tratava também de um doutor impedindo essa acção enviando a mensagem de erro apresentada na imagem 18. Quando se tentou realizar um update de um doutor usando o VAT de uma recepcionista, o trigger também não permitiu a concretização da tarefa.

```

MySQL [ist187124]> INSERT INTO doctor
-> VALUES (10202, 'oral medicine', 'Graduated with honors.', 'sweet.jane@gmail.com');
ERROR 1644 (45000): IC: person VAT 10202 is already a nurse, cannot be a doctor too.
MySQL [ist187124]>
MySQL [ist187124]> INSERT INTO doctor
-> VALUES (10401, 'oral medicine', 'Graduated with honors.', 'sweet.jane@gmail.com');
ERROR 1644 (45000): IC: person VAT 10401 is already a recepcionist, cannot be a doctor too.
MySQL [ist187124]>
MySQL [ist187124]> UPDATE doctor SET VAT = 10401 WHERE VAT = 10051;
ERROR 1644 (45000): IC: person VAT 10401 is already a recepcionist, cannot be a doctor too.

```

Figura 18: Testing task 2a

4.2.2 b)

Um doutor que seja permanente não poder ser um trainee simultaneamente e vice-versa. Para tal, executa-se um procedimento semelhante à alínea anterior, ou seja, verificar se antes de uma inserção ou uma atualização a uma das tabelas do doutor se este não existe já na outra tabela. O código realizado para a execução da Task 2b) encontra-se no *listing* 8.

Listing 8: Task2

```

1  /*procedure to check if a trainee doctor is not in the permanent_doctor
   table*/
2  CREATE PROCEDURE check_doct_perm
3  (IN VAT INTEGER)
4  BEGIN
5      DECLARE msg VARCHAR(255);
6      IF EXISTS (SELECT permanent_doctor.VAT FROM permanent_doctor
7                  WHERE permanent_doctor.VAT = VAT) THEN
8          SET msg = CONCAT("IC: person VAT ", VAT,
9                          " is already a permanent doctor, cannot be a trainee doctor too
10                         .");
11
12          SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
13      END IF;
14  END$$
15 /*triggers to call procedure to check alterations in trainee_doctor
   table*/
16 CREATE TRIGGER ins_trainee BEFORE INSERT ON trainee_doctor
17 FOR EACH ROW
18 BEGIN
19     CALL check_doct_perm(NEW.VAT);
20 END$$
21 CREATE TRIGGER upd_trainee BEFORE UPDATE ON trainee_doctor
22 FOR EACH ROW
23 BEGIN
24     CALL check_doct_perm(NEW.VAT);
25 END$$
26 /*procedure to check if a permanent doctor is not in the trainee_doctor
   table*/
27 CREATE PROCEDURE check_doct_trainee
28 (IN VAT INTEGER)
29 BEGIN
30     DECLARE msg VARCHAR(255);
31     IF EXISTS (SELECT trainee_doctor.VAT FROM trainee_doctor
32                 WHERE trainee_doctor.VAT = VAT) THEN
33         SET msg = CONCAT("IC: person VAT ", VAT,
34                         " is already a trainee doctor, cannot be a permanent doctor too
35                        .");
36
37         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
38     END IF;
39 END$$
40 /*triggers to call procedure to check alterations in permanent_doctor
   table*/
41 CREATE TRIGGER ins_perm BEFORE INSERT ON permanent_doctor
42 FOR EACH ROW
43 BEGIN

```

```

42     CALL check_doct_trainee(NEW.VAT);
43 END$$
44 CREATE TRIGGER upd_perm BEFORE UPDATE ON permanent_doctor
45 FOR EACH ROW
46 BEGIN
47     CALL check_doct_trainee(NEW.VAT);
48 END

```

Foram então criadas duas procedures que cada uma tem como argumento o VAT inserido na tentativa de inserção ou atualização de um dado de uma tabela. Cada procedure trata de uma verificação, ou dos doutores permanentes ou dos doutores trainees. Estas duas procedures, através da criação dos respetivos triggers, são chamadas sempre antes de uma inserção e de uma atualização através dos comandos *BEFORE INSERT ON* e *BEFORE UPDATE ON*.

Para testar a segunda task, foram desenvolvidas umas queries apresentadas no *listing* 9.

Listing 9: TestTask2

```

1 INSERT INTO trainee_doctor
2 VALUES (10002, 10050);
3 UPDATE trainee_doctor SET VAT = 10001 WHERE VAT = 10050;

```

Tentou-se inserir e atualizar na tabela dos doutores trainees um doutor permanente que o trigger detetou que se tratava de um doutor permanente e enviou a mensagem de erro apresentada na imagem 19.

```

MySQL [ist187124]> /* 2. Test trigger doctors and trainee */
MySQL [ist187124]> INSERT INTO trainee_doctor
-> VALUES (10002, 10050);
ERROR 1644 (45000): IC: person VAT 10002 is already a permanent doctor, cannot be a trainee doctor too.
MySQL [ist187124]>
MySQL [ist187124]> UPDATE trainee_doctor
-> SET VAT = 10001 WHERE VAT = 10050;
ERROR 1644 (45000): IC: person VAT 10001 is already a permanent doctor, cannot be a trainee doctor too.

```

Figura 19: Testing task 2b

4.3 Task 3

Nesta terceira Task, é pedido para que seja assegurado que diferentes pessoas (clientes ou doutores) tenham necessariamente números de telefone diferentes. Assumiu-se que os clientes têm de ter números de telefone diferentes de qualquer funcionário e também diferentes de outro cliente e o mesmo se sucede para o caso de um funcionário, que terá de ter um número de telefone diferente de todos os outros funcionários e clientes. O código desenvolvido para a task 3 encontra-se no *listing* 10.

Listing 10: Task3

```

1 CREATE TRIGGER ins_phone_cl BEFORE INSERT ON phone_number_client
2 FOR EACH ROW
3 BEGIN
4     DECLARE msg VARCHAR(255);
5     IF EXISTS (SELECT phone_number_client.phone FROM
phone_number_client

```

```

6         WHERE phone_number_client.phone = NEW.phone) THEN
7     SET msg = CONCAT(IC: There s a client with that phone_number ,
8         NEW.phone ,  ".");
9
10    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
11 END IF;
12 IF EXISTS (SELECT phone_number_employee.phone FROM
phone_number_employee
13         WHERE phone_number_employee.phone = NEW.phone) THEN
14     SET msg = CONCAT(IC: There s an employee with that phone_number
15 ,
16         NEW.phone ,  ".");
17
18    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
19 END IF;
20 END$$
21 CREATE TRIGGER ins_phone_emp BEFORE INSERT ON phone_number_employee
22 FOR EACH ROW
23 BEGIN
24     DECLARE msg VARCHAR(255);
25     IF EXISTS (SELECT phone_number_client.phone FROM
phone_number_client
26         WHERE phone_number_client.phone = NEW.phone) THEN
27         SET msg = CONCAT(IC: There s a client with that phone_number ,
28             NEW.phone ,  ".");
29
30     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
31 END IF;
32 IF EXISTS (SELECT phone_number_employee.phone FROM
phone_number_employee
33         WHERE phone_number_employee.phone = NEW.phone) THEN
34     SET msg = CONCAT(IC: Theres an employee with that phone_number
35 ,
36         NEW.phone ,  ".");
37
38    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
39 END IF;
40 END$$
41 CREATE TRIGGER upd_phone_cl BEFORE UPDATE ON phone_number_client
42 FOR EACH ROW
43 BEGIN
44     DECLARE msg VARCHAR(255);
45     IF EXISTS (SELECT phone_number_client.phone FROM
phone_number_client
46         WHERE phone_number_client.phone = NEW.phone) THEN
47         SET msg = CONCAT(IC: There s a client with that phone_number ,
48             NEW.phone ,  ".");
49
50     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
51 END IF;
52 IF EXISTS (SELECT phone_number_employee.phone FROM
phone_number_employee
53         WHERE phone_number_employee.phone = NEW.phone) THEN
54     SET msg = CONCAT(IC: There s an employee with that phone_number
55 ,
56         NEW.phone ,  ".");
57
58    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;

```



```

56     END IF;
57 END$$
58 CREATE TRIGGER upd_phone_emp BEFORE UPDATE ON phone_number_employee
59 FOR EACH ROW
60 BEGIN
61     DECLARE msg VARCHAR(255);
62     IF EXISTS (SELECT phone_number_client.phone FROM
63         phone_number_client
64         WHERE phone_number_client.phone = NEW.phone) THEN
65         SET msg = CONCAT(IC: There s a client with that phone_number ,
66             NEW.phone , ".");
67     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
68 END IF;
69 IF EXISTS (SELECT phone_number_employee.phone FROM
70     phone_number_employee
71     WHERE phone_number_employee.phone = NEW.phone) THEN
72     SET msg = CONCAT(IC: There s an employee with that phone_number
73         ,
74         NEW.phone , ".");
75     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = msg;
76 END IF;
77 END$$

```

Estas triggers criadas para a execução da terceira task são disparadas antes de cada inserção ou atualização de um número de telefone, quer seja no de um cliente ou num de um employee.

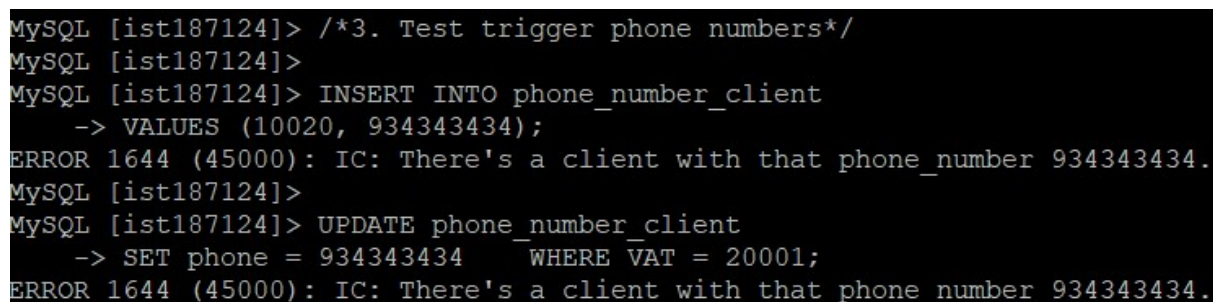
Para testar a terceira task, foram desenvolvidas umas queries apresentadas no *listing* 11.

Listing 11: TestTask3

```

1 /*3. Test trigger phone numbers*/
2 INSERT INTO phone_number_client
3 VALUES (10020, 934343434);
4 UPDATE phone_number_client SET phone = 934343434 WHERE VAT =
    20001;

```



```

MySQL [ist187124]> /*3. Test trigger phone numbers*/
MySQL [ist187124]>
MySQL [ist187124]> INSERT INTO phone_number_client
-> VALUES (10020, 934343434);
ERROR 1644 (45000): IC: There's a client with that phone_number 934343434.
MySQL [ist187124]>
MySQL [ist187124]> UPDATE phone_number_client
-> SET phone = 934343434 WHERE VAT = 20001;
ERROR 1644 (45000): IC: There's a client with that phone_number 934343434.

```

Figura 20: Testing task 3

Verifica-se na imagem 20 que ao se tentar inserir um número de telefone de um doutor na tabela dos números de telefone dos clientes, o trigger dispara e mostra a mensagem de erro apresentada. O mesmo acontece para a atualização que se tentou fazer.

4.4 Task 4

Na 4ª tarefa é pedido para criar uma função que calcule o número total de marcações de consultas a que as pessoas faltaram, conforme o género do cliente, a sua idade, e o ano de marcação da consulta.

Para tal são seleccionados os clientes que se enquadram no caso escolhido, e são contadas as *appointments* em que estes participam, que não se tornaram consultas. O código desta função encontra-se no *listing* 12.

Listing 12: Função da tarefa 4

```

1 CREATE FUNCTION number_of_no_shows(c_gender VARCHAR(8), min_age INTEGER
  , max_age INTEGER, year INTEGER)
2 RETURNS INTEGER
3 BEGIN
4   DECLARE ns_total INTEGER;
5   SELECT COUNT(*) INTO ns_total
6   FROM (SELECT ap.date_timestamp as dates, ap.VAT_doctor as vat_doc, ap
  .VAT_client as vat_client, c.s as soap
7         FROM appointment as ap
8         LEFT OUTER JOIN consultation as c
9         ON ap.VAT_doctor=c.VAT_doctor and ap.date_timestamp= c.
  date_timestamp
10        WHERE YEAR(ap.date_timestamp) = year
11        AND c.s IS NULL
12        AND vat_client IN(SELECT VAT_client
13                          FROM client
14                          WHERE gender = c_gender AND age > min_age AND
  age < max_age))as apc;
15   RETURN ns_total;
16 END $$

```

Esta função é chamada quando é necessário utilizar como se mostra no *listing* 13.

Listing 13: Chamada da função criada

```

1 SELECT number_of_no_shows('Male', 0, 100, 2019);

```

4.5 Task 5

Para esta tarefa foi pedido para criar um procedimento que alterasse o salário dos médicos que trabalhavam há mais de x anos, sendo x o valor escolhido ao chamar o procedimento. Para o caso dos médicos que estivessem presentes em mais de 100 consultas no ano corrente, o seu salário seria aumentado 10%, para os restantes, seria aumentado 5%.

Para tal, procurou-se os médicos que exercem há mais de x anos e em quantas consultas estes participaram no ano corrente. O seus salários foram atualizados com a função *update*, segundo as instruções dadas. O código do procedimento encontra-se no *listing* 14.

Listing 14: Task 5

```

1 CREATE PROCEDURE change_salary(in x INTEGER)
2 BEGIN

```

```

3  UPDATE employee
4  SET salary = salary*1.10
5  WHERE VAT IN (SELECT s.VAT_doctor
6                 FROM (SELECT c.VAT_doctor, COUNT(*) AS apps
7                       FROM consultation AS c
8                       WHERE YEAR(c.date_timestamp) = YEAR(CURDATE()))
9                 GROUP BY c.VAT_doctor
10                ) AS s
11                WHERE s.apps > 100)
12  AND VAT IN (SELECT VAT
13              FROM permanent_doctor
14              WHERE years > x);
15
16  UPDATE employee
17  SET salary = salary*1.05
18  WHERE VAT IN (SELECT s.VAT_doctor
19                 FROM (SELECT c.VAT_doctor, COUNT(*) AS apps
20                       FROM consultation AS c
21                       WHERE YEAR(c.date_timestamp) = YEAR(CURDATE()))
22                 GROUP BY c.VAT_doctor
23                ) AS s
24                WHERE s.apps <= 100)
25  AND VAT IN (SELECT VAT
26              FROM permanent_doctor
27              WHERE years > x);
28  END $$

```

Para testar este procedimento foi utilizado o seguinte código, no *listing* 15, que mostra o salário de cada médico, há quanto tempo estes praticam, e quantas consultas deram no ano corrente, antes e depois do procedimento ser chamado.

Listing 15: Task 5

```

1  SELECT employee.VAT, name, salary, years, Nm_of_Consultations
2  FROM employee
3  INNER JOIN permanent_doctor ON employee.VAT = permanent_doctor.VAT
4  INNER JOIN (SELECT c.VAT_doctor, COUNT(*) AS Nm_of_Consultations
5               FROM consultation AS c
6               WHERE YEAR(c.date_timestamp) = YEAR(CURDATE())
7               GROUP BY c.VAT_doctor) AS s
8           ON employee.VAT = s.VAT_doctor;
9
10 CALL change_salary(17);
11
12 SELECT employee.VAT, name, salary, years, Nm_of_Consultations
13 FROM employee
14 INNER JOIN permanent_doctor ON employee.VAT = permanent_doctor.VAT
15 INNER JOIN (SELECT c.VAT_doctor, COUNT(*) AS Nm_of_Consultations
16              FROM consultation AS c
17              WHERE YEAR(c.date_timestamp) = YEAR(CURDATE())
18              GROUP BY c.VAT_doctor) AS s
19           ON employee.VAT = s.VAT_doctor;

```

Como se pode observar na figura 21, apenas o salário do médico que trabalha há mais de 17 anos é alterado, e dado que este médico não deu mais de 100 consultas no ano corrente, este é apenas aumentado 5%.

```

+-----+-----+-----+-----+-----+
| VAT   | name           | salary | years | Nm_of_Consultations |
+-----+-----+-----+-----+-----+
| 10001 | Ada Lovelace   | 1500.00 | 15 | 6 |
| 10002 | Jane Sweettooth | 1600.00 | 20 | 5 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+-----+-----+-----+-----+-----+
| VAT   | name           | salary | years | Nm_of_Consultations |
+-----+-----+-----+-----+-----+
| 10001 | Ada Lovelace   | 1500.00 | 15 | 6 |
| 10002 | Jane Sweettooth | 1680.00 | 20 | 5 |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

```

Figura 21: Teste do procedimento.

5 Conclusão

Este terceiro projeto permitiu entender como se pode manipular uma base de dados através de uma aplicação Web. Assim, foi possível aplicar os conhecimentos adquiridos nas aulas teóricas e de laboratório em relação à criação dos scripts em PHP e HTML de modo a poder cumprir as tarefas do enunciado do projeto. As páginas WEB foram criadas com sucesso e entendeu-se a importância da passagem de variáveis de página para página e o modo como é realizada a interação entre estas páginas WEB e a base de dados. Compreendeu-se o uso das funções, dos triggers e das procedures e as suas funcionalidades quando são aplicados uma vez que simplificam processos repetitivos, permitem evitar erros quer na inserção, quer na atualização de dados na base de dados. Estes podem ser chamados quando se deteta alguma modificação na base de dados numa qualquer tabela indicada. Permitem ainda um acesso mais simples e intuitivo aos dados bastando por vezes chamar uma função ou um procedure em vez de se utilizar a mesma query várias vezes.

6 Anexos

6.1 Inicial Page

6.1.1 inicial_page.php

Welcome to the Database Search Tool:

What do you wish to do?

Figura 22: Página Inicial

Listing 16: Codigo inicial_page.php

```
1 <html>
2 <body><h2>Welcome to the Database Search Tool:</h2>
3   <h3>What do you wish to do?</h3>
4
5   <p><form action='search_client_form.php' method='post'>
6     <input type='submit' value='Search For Clients' />
7   </form></p>
8   <p><form action='new_client_form.php' method='post'>
9     <input type='submit' value='Add a Client' />
10  </form></p>
11 </body>
12 </html>
```

6.2 Add clients

6.2.1 new_client_form.php

Add Client:

VAT:

Name:

Birthdate:

Street:

City:

ZIP:

Phone Number:

Gender:

Figura 23: Novo Cliente

Listing 17: Codigo new_client_form.php

```
1 <html>
2 <body><h2>Add Client:</h2>
3   <form action="new_client.php" method="post">
4     <p>VAT:
5       <input type="number" min="0" name="VAT_client" required/>
6     </p>
7     <p>Name:
8       <input type="text" name="client_name" maxlength='39'
required/>
9     </p>
10    <p>Birthdate:
11      <input type="date" name="client_birthdate" maxlength='34'
required/>
12    </p>
13    <p>Street:
14      <input type="text" name="client_street" maxlength='34'
required/>
15    </p>
16    <p>City:
17      <input type="text" name="client_city" maxlength='34'
required/>
```

```

18     </p>
19     <p>ZIP:
20         <input type="text" name="client_zip" maxlength='34'
required/>
21     </p>
22     <p>Phone Number:
23         <input type="number" name="client_phone" maxlength='34'
required/>
24     </p>
25     <p>Gender:
26         <select name="client_gender">
27             <option value="Male">Male</option>
28             <option value="Female">Female</option>
29         </select>
30     </p>
31     <p></p>
32     <p><input type="submit" value="Add"/></p>
33 </form>
34     <a href="javascript:history.go(-1)">
35         <input type='submit' value='Go Back' /></a>
36 </form>
37 <p></p>
38 <form action='initial_page.php' method='post'>
39 <input type='submit' value='Go to Homepage' />
40 </form>
41 </body>
42 </html>

```

6.2.2 and new_client.php

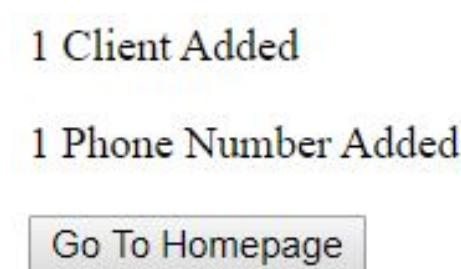


Figura 24: Confirmação do cliente

Listing 18: Código new_client.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5         $VAT_client = $_REQUEST['VAT_client'];
6         $client_name = $_REQUEST['client_name'];
7         $client_birthdate = $_REQUEST['client_birthdate'];
8         $client_street = $_REQUEST['client_street'];
9         $client_city = $_REQUEST['client_city'];

```

```
10     $client_zip = $_REQUEST['client_zip'];
11     $client_gender = $_REQUEST['client_gender'];
12     $client_phone = $_REQUEST['client_phone'];
13
14     # TRY TO CONNECT WITH DB
15     $host = "db.tecnico.ulisboa.pt";
16     $user = "ist425487";
17     $pass = "vief8835";
18     $dsn = "mysql:host=$host;dbname=$user";
19     $connection = null;
20     try
21     {
22         $connection = new PDO($dsn, $user, $pass);
23     }
24     catch(PDOException $exception)
25     {
26         echo("<p>Error: ");
27         echo($exception->getMessage());
28         echo("</p>");
29         exit();
30     }
31
32     $today = date("Y-m-d");//end time
33     $client_birthdate=date('Y-m-d', strtotime("$client_birthdate"))
34 ;
35     $diff = abs(strtotime($today) - strtotime($client_birthdate));
36     $age = floor($diff / (365*60*60*24));
37
38     $sql_VAT_check= "SELECT * FROM client WHERE VAT_client=:
39 VAT_client";
40     $check_vat = $connection->prepare($sql_VAT_check);
41     $result=$check_vat ->execute([
42         'VAT_client' => $VAT_client,
43     ]);
44     if ($result== FALSE)
45     {
46         echo("Add client query.");
47         $info = $connection->errorInfo();
48         echo("<p>Error: {$info[1]}</p>");
49         exit();
50     }
51     $n_vat=$check_vat->rowCount();
52     if($n_vat==0)
53     {
54         $sql1 = "INSERT INTO client VALUES
55             ($VAT_client, '$client_name', '$client_birthdate', '
56 $client_street', '$client_city', '$client_zip', '$client_gender',
57 $age)";
58
59         $sql2 = "INSERT INTO phone_number_client VALUES (
60 $VAT_client, $client_phone)";
61
62         $nrows = $connection->exec($sql1);
63         if($nrows>0)
64         {
65             echo("<p>$nrows Client Added</p>");
66         }
67         elseif ($nrows== FALSE)
```

```
63     {
64         echo("<p>Unable to add client.</p>");
65         $info = $connection->errorInfo();
66         echo("<p>Error: {$info[1]}</p>");
67         exit();
68     }
69
70     $nrows = $connection->exec($sql2);
71     if($nrows>0)
72     {
73         echo("<p>$nrows Phone Number Added</p>");
74     }
75     elseif ($nrows== FALSE)
76     {
77         echo("<p>Unable to add client phone number.</p>");
78         $info = $connection->errorInfo();
79         echo("<p>Error: {$info[1]}</p>");
80         exit();
81     }
82 }else{
83     echo("<p>VAT for client already used</p>");
84     $sql2 = "INSERT INTO phone_number_client VALUES (
85 $VAT_client, $client_phone)";
86     $nrows = $connection->exec($sql2);
87     if($nrows>0)
88     {
89         echo("<p>$nrows Phone Number Added</p>");
90     }
91     elseif ($nrows== FALSE)
92     {
93         echo("<p>Unable to add client phone number.</p>");
94         $info = $connection->errorInfo();
95         echo("<p>Error: {$info[1]}</p>");
96         exit();
97     }
98 }
99 echo("<form action='initial_page.php' method='post'>
100 <input type='submit' value='Go To Homepage' />
101 </form>");
102
103 $connection = null;
104
105 </body>
106 </html>
```


6.3 Search for and display Clients

6.3.1 search_client_form.php

Search Client:

Client VAT:

Client Name:

Client Address:

Figura 25: Procura por Cliente

Listing 19: Código search_client_form.php

```
1 <html>
2 <body><h2>Search Client:</h2>
3     <form action="search_client.php" method="post">
4         <p>Client VAT:
5             <input type="number" min="0" name="VAT_client" />
6         </p>
7         <p>Client Name:
8             <input type="text" name="client_name" maxlength='39' />
9         </p>
10        <p>Client Address:
11            <input type="text" name="client_addr" maxlength='34' />
12        </p>
13        <p><input type="submit" value="Search"/></p>
14    </form>
15    <form action='new_client_form.php' method='post'>
16        <input type='submit' value='Add new client' />
17    </form>
18    <form action='initial_page.php' method='post'>
19        <input type='submit' value='Go To Homepage' />
20    </form>
21 </body>
22 </html>
```

6.3.2 search_client.php

Client:

Number of Clients: 9

VAT_client	Name	Address	City	Zip	Appointment Records	Add Appointment
1	fghjk	Avenida das Forcas Armadas n°133 1	Lisbon	1600-081	here	here
20001	Jane Doe	Yasmine Road	New Jersey	20101	here	here
20002	Merida of DunBroch	Toad Road	New Yorkshire	20102	here	here
20003	Maui	Ocean Street	Pacific Island	20103	here	here
20004	Spider Man	Brooklyn Street	New York City	20104	here	here
20005	Walter White	308 Negra Arroyo Lane	Albuquerque	20105	here	here
20006	Hermione Granger	Gryffindor Tower	Hogwarts	20106	here	here
20007	Hiccup Horrendous Haddock	Viking Lane	Island	20107	here	here
20008	Eleanor Shellstrop	The Good Place Lane	Afterlife	20108	here	here

[Go Back](#)

[Add New Client](#)

Figura 26: Lista de Clientes

Listing 20: Codigo search_client.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5         $VAT_client = (integer)$_REQUEST['VAT_client'];
6         $client_name = (string)$_REQUEST['client_name'];
7         $client_addr = (string)$_REQUEST['client_addr'];
8
9         # TRY TO CONNECT WITH DB
10        $host = "db.tecnico.ulisboa.pt";
11        $user = "ist425487";
12        $pass = "vief8835";
13        $dsn = "mysql:host=$host;dbname=$user";
14        $connection = null;
15        try
16        {
17            $connection = new PDO($dsn, $user, $pass);
18        }
19        catch(PDOException $exception)
20        {
21            echo("<p>Error: ");
22            echo($exception->getMessage());
23            echo("</p>");
24            exit();
25        }
26        #
27        #####
28
29        if(!empty($VAT_client))
30        {
31            #####
32            if(empty($client_name) && empty($client_addr)) //
33                quando existe um num VAT no form

```

```

31         {
32             $find_clients = $connection->prepare("SELECT *
33                                     FROM client
34                                     WHERE VAT_client =:
VAT_client");
35
36             $result=$find_clients ->execute([
37                 'VAT_client' => $VAT_client,
38             ]);
39             if ($result== FALSE)
40             {
41                 echo("Client search query(1)");
42                 $info = $connection->errorInfo();
43                 echo("<p>Error: {$info[1]}</p>");
44                 exit();
45             }
46             #####
47             }elseif(!empty($client_name) && empty($client_addr)) //
quando existe um num VAT e um nome no form
48             {
49                 $find_clients = $connection->prepare("SELECT *
50                                     FROM client
51                                     WHERE VAT_client =:
VAT_client
52                                     AND name LIKE CONCAT('%',:
client_name,'%')");
53
54                 $result=$find_clients ->execute([
55                     'VAT_client' => $VAT_client,
56                     'client_name' => $client_name,
57                 ]);
58                 if ($result== FALSE)
59                 {
60                     echo("Client search query(2)");
61                     $info = $connection->errorInfo();
62                     echo("<p>Error: {$info[1]}</p>");
63                     exit();
64                 }
65                 #####
66                 }elseif(empty($client_name) && !empty($client_addr)){ //
quando existe um num VAT e uma morada no form
67                 $find_clients = $connection->prepare("SELECT *
68                                     FROM client
69                                     WHERE VAT_client =:
VAT_client
70                                     AND street LIKE
CONCAT('%',:client_addr,'%') OR zip LIKE CONCAT('%',:client_addr
,'%') OR city LIKE CONCAT('%',:client_addr,'%')");
71
72                 $result=$find_clients ->execute([
73                     'VAT_client' => $VAT_client,
74                     'client_addr' => $client_addr,
75                 ]);
76                 if ($result== FALSE)
77                 {
78                     echo("Client search query(3)");
79                     $info = $connection->errorInfo();
80                     echo("<p>Error: {$info[1]}</p>");

```

```

81         exit();
82     }
83     #####
84     }elseif(!empty($client_name) && !empty($client_addr)){ //
quando existe um num VAT, um nome e uma morada no form
85         $find_clients = $connection->prepare("SELECT *
86                                             FROM client
87                                             WHERE VAT_client =:
VAT_client
88                                             AND name LIKE
CONCAT('%',:client_name,'%')
89                                             AND street LIKE
CONCAT('%',:client_addr,'%') OR zip LIKE CONCAT('%',:client_addr
,'%') OR city LIKE CONCAT('%',:client_addr,'%')");
90
91         $result=$find_clients ->execute([
92             'VAT_client' => $VAT_client,
93             'client_name' => $client_name,
94             'client_addr' => $client_addr,
95         ]);
96         if ($result== FALSE)
97         {
98             echo("Client search query(4)");
99             $info = $connection->errorInfo();
100            echo("<p>Error: {$info[1]}</p>");
101            exit();
102        }
103    }
104 }
105 #
#####
106 elseif(empty($VAT_client) && !empty($client_name))
107 {
108     if(!empty($client_addr)){
109
110         $find_clients = $connection->prepare("SELECT *
111                                             FROM client
112                                             WHERE client.name LIKE
CONCAT('%',:client_name,'%')
113                                             AND street LIKE CONCAT
('%',:client_addr,'%') OR zip LIKE CONCAT('%',:client_addr,'%') OR
city LIKE CONCAT('%',:client_addr,'%')");
114
115         $result=$find_clients ->execute([
116             'client_name' => $client_name,
117             'client_addr' => $client_addr,
118         ]);
119         if ($result== FALSE)
120         {
121             echo("Client search query(5)");
122             $info = $connection->errorInfo();
123             echo("<p>Error: {$info[1]}</p>");
124             exit();
125         }
126     }elseif{
127         $find_clients = $connection->prepare("SELECT *
128                                             FROM client

```

```

129                                                                 WHERE client.name
130     LIKE CONCAT('%',:client_name,'%')");
131
132     $result=$find_clients ->execute([
133         'client_name' => $client_name,
134     ]);
135     if ($result== FALSE)
136     {
137         echo("Client search query(6)");
138         $info = $connection->errorInfo();
139         echo("<p>Error: {$info[1]}</p>");
140         exit();
141     }
142 }
143 #
144 #####
145
146 elseif(empty($VAT_client) && empty($client_name) && !empty(
147 $client_addr))
148 {
149     $find_clients = $connection->prepare("SELECT *
150                                         FROM client
151                                         WHERE client.street LIKE
152 CONCAT('%',:client_addr,'%') OR client.zip LIKE CONCAT('%',:
153 client_addr,'%') OR client.city LIKE CONCAT('%',:client_addr,'%')");
154
155     $result=$find_clients ->execute([
156         'client_addr' => $client_addr,
157     ]);
158     if ($result== FALSE)
159     {
160         echo("Client search query(7)");
161         $info = $connection->errorInfo();
162         echo("<p>Error: {$info[1]}</p>");
163         exit();
164     }
165 }
166 #
167 #####
168
169 elseif(empty($VAT_client) && empty($client_name) && empty(
170 $client_addr))
171 {
172     $find_clients = $connection->prepare("SELECT *
173                                         FROM client");
174
175     $result=$find_clients ->execute();
176     if ($result== FALSE)
177     {
178         echo("Client search query(8)");
179         $info = $connection->errorInfo();
180         echo("<p>Error: {$info[1]}</p>");
181         exit();
182     }
183 }

```

```

178     $n_clients=$find_clients->rowCount();
179     if($n_clients>0)
180     {
181         echo("<h4>Client:</h4>");
182
183         echo("<p> Number of Clients: {$n_clients}</p>" );
184         echo("<table border=\"1\" style=\"margin-bottom: 20px;\"> "
185 );
186         echo("<tr>");
187         echo("<td>VAT_client</td>");
188         echo("<td>Name</td>");
189         echo("<td>Address</td>");
190         echo("<td>City</td>");
191         echo("<td>Zip</td>");
192         echo("<td>Appointment Records</td>");
193         echo("<td>Add Appointment</td>");
194         echo("</tr>\n");
195         {
196             while($client=$find_clients->fetch()){
197                 echo("<tr>");
198
199                 echo("<td>{$client['VAT_client']}'</td>");
200                 echo("<td>{$client['name']}'</td>");
201                 echo("<td>{$client['street']}'</td>");
202                 echo("<td>{$client['city']}'</td>");
203                 echo("<td>{$client['zip']}'</td>");
204                 echo("<td><a href=\"apps_and_cons.php?VAT_client=\"");
205
206                 echo($client['VAT_client']);
207                 echo(">here</a></td>\n");
208                 echo("<td><a href=\"add_app_form.php?VAT_client=\"");
209                 echo($client['VAT_client']);
210                 echo(">here</a></td>\n");
211                 echo("</tr>\n");
212             }
213         }
214         #
215         #####
216
217         ###          QUANDO SE CARREGAR NO CLIENTE
218         ###
219         ###          action='apps_and_cons.php' (ainda nao esta feita)
220         ###
221         #
222         #####
223
224         echo("</table>");
225         echo ("<a href=\"javascript:history.go(-1)\">
226         <input type='submit' value='Go Back' /></a>
227         </form>");
228         echo("<form action='new_client_form.php' method='post'>
229         <input type='submit' value='Add New Client' />
230         </form>");
231     }else

```

```

228     {
229         echo("<p>Client not found</p>\n");
230         echo ("<a href=\"javascript:history.go(-1)\">
231             <input type='submit' value='Go Back' /></a>
232             </form>");
233     }
234
235     # END CONNECTION
236     $connection = null;
237 ?>
238
239 </body>
240 </html>

```

6.4 Appointments and Consultations

6.4.1 apps_and_cons.php

Client: VAT= 20001 and Name= Jane Doe

Appointments and Consultations:

Number of Appointments/Consultations: 4

Appointment Date	Status	Link
2019-03-27 11:00:00	Consultation	More Information
2019-12-04 11:00:00	Consultation	More Information
2019-12-04 13:00:00	Passed Appointment	More Information
2019-12-11 10:00:00	Appointment booked	More Information

Add New Appointment

Go Back to Search

Figura 27: Lista das Appointments de um Cliente

Listing 21: Codigo apps_and_cons.php

```

1 <html>
2 <body>
3     <?php
4         #FORM
5         $VAT_client = (integer)$_REQUEST['VAT_client'];
6         # TRY TO CONNECT WITH DB
7         $host = "db.tecnico.ulisboa.pt";

```

```

8      $user = "ist425487";
9      $pass = "vief8835";
10     $dsn = "mysql:host=$host;dbname=$user";
11     $connection = null;
12     try
13     {
14         $connection = new PDO($dsn, $user, $pass);
15     }
16     catch(PDOException $exception)
17     {
18         echo("<p>Error: ");
19         echo($exception->getMessage());
20         echo("</p>");
21         exit();
22     }
23     $find_name = $connection->prepare("SELECT name
24                                     FROM client
25                                     WHERE VAT_client = :
VAT_client");
26     $result=$find_name->execute(['VAT_client' => $VAT_client]);
27     $n_client = $find_name->rowCount();
28     if ($result== FALSE)
29     {
30         echo("Couldn't find client");
31         $info = $connection->errorInfo();
32         echo("<p>Error: {$info[1]}</p>");
33         exit();
34     }
35     $name=$find_name->fetchAll();
36     foreach($name as $row){
37         $client_name=$row['name'];
38     }
39     echo("<h2>Client: VAT= {$VAT_client} and Name= {$client_name}</
h2>");
40
41     $find_apps = $connection->prepare("SELECT ap.date_timestamp as
dates, ap.VAT_doctor as VAT_doc, c.s as s, c.o as o, c.a as a, c.p
as p
                                     FROM appointment as ap
                                     LEFT OUTER JOIN consultation
as c
                                     ON ap.VAT_doctor=c.
VAT_doctor and ap.date_timestamp= c.date_timestamp
                                     WHERE ap.VAT_client = :
VAT_client
                                     ORDER BY ap.date_timestamp
ASC");
42
43
44
45
46
47
48     $result=$find_apps->execute(['VAT_client' => $VAT_client]);
49     $n_apps = $find_apps->rowCount();
50     if($n_apps>0)
51     {
52         echo("<h3>Appointments and Consultations:</h3>");
53         echo("<p> Number of Appointments/Consultations: {$n_apps}</
p>");
54         echo("<table border=\"1\" style=\"margin-bottom: 20px;\"> ");
55         echo("<tr>");

```



```

56         echo("<td>Appointment Date</td>");
57         echo("<td>Status</td>");
58         echo("<td>Link</td>");
59         echo("</tr>\n");
60     {
61         while($app=$find_apps->fetch()){
62             echo("<tr>");
63             echo("<td>{$app['dates']}</td>");
64             if($app['dates']>date("Y-m-d")){
65                 echo("<td>Appointment booked</td>");
66                 $isConsult="booked";
67             }elseif($app['s']==NULL){
68                 echo("<td>Passed Appointment</td>");
69                 $isConsult="missed";
70             }else{
71                 echo("<td>Consultation</td>");
72                 $isConsult="consultation";
73             }
74             echo("<td><a href=\"list_cons_info.php?
date_timestamp=\">");
75             echo($app['dates']);
76             echo("&VAT_doctor=");
77             echo($app['VAT_doc']);
78             echo("&isConsult=");
79             echo($isConsult);
80             echo("&VAT_client=");
81             echo($VAT_client);
82             echo("</a></td>\n");
83             echo("</tr>\n");
84         }
85     }
86     echo("</table>");
87 }else{
88     echo("<h4>No Appointments nor Consultations found</h4>"
);
89 }
90
91 #
#####
92
93     ###          PROCURAR CONSULTAS/APPS DO CLIENTE
94     ###
95     ###          QUANDO SE CARREGAR NA CONSULTA
96     ###
97     ###          action='list_cons_info.php' (ainda nao esta feita)
98     ###
99     #
100 #####
101
102     echo("<form action='add_app_form.php' method='post'>
103     <input type='submit' value='Add New Appointment' />
104     <input type='hidden' name='VAT_client' value='{ $VAT_client }'>
105     </form>");
106     echo("<form action='search_client_form.php' method='post'>
107     <input type='submit' value='Go Back to Search' />
108     </form>");

```

```
105
106
107     $connection = null;
108     ?>
109
110 </body>
111 </html>
```

6.4.2 list_cons_info.php

Appointment Information:

Date and Time	Doctor's name	Doctor's VAT	Description
2019-12-17 09:00:00	Jane Sweettooth	10002	Regular Check up

Go Back

Figura 28: Informação de uma Appointment

Appointment Information:

Date and Time	Doctor's name	Doctor's VAT	Description
2019-08-15 11:00:00	Ada Lovelace	10001	To come check in.

Add Information

Add Dental Charting Informations

Go Back

Figura 29: Informação de uma Appointment que já passou

Consultation Information:

VAT_client	Consultation Date and Time	Doctor's Name	Doctor's VAT	Appointment Description
20002	2018-02-27 11:00:00	Ada Lovelace	10001	To eliminate rotting tooth.

Nurse

Nurse name	Nurse VAT
10201	Waldo Missing

SOAP:

Subjective Observation	Objective Observation	Assessment	Plan
Client is redhead.	Has nothing else wrong.	Doing well	Recommend no sugar.

Diagnosis:

Diagnostic	Description
80967002	dental cavities.

Prescriptions:

Prescription	Lab	Dosage	Description
ibuprofen	Ibu Lab	2	Take one every 8 hours.

Add Dental Charting Informations

Go Back

Figura 30: Informação de uma Consulta

Listing 22: Código list_cons_info.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5         header("refresh: 60;");
6         $date = strtotime($_REQUEST['date_timestamp']);
7         $date_timestamp=date('Y-m-d H:i:s', $date);
8
9         $VAT_doctor = (integer)$_REQUEST['VAT_doctor'];
10        $isConsult = (string)$_REQUEST['isConsult'];
11        $VAT_client = (integer)$_REQUEST['VAT_client'];
12
13        # TRY TO CONNECT WITH DB
14        $host = "db.tecnico.ulisboa.pt";
15        $user = "ist425487";
16        $pass = "vief8835";
17        $dsn = "mysql:host=$host;dbname=$user";
18        $connection = null;
19        try
20        {
21            $connection = new PDO($dsn, $user, $pass);

```

```

22     }
23     catch(PDOException $exception)
24     {
25         echo("<p>Error: ");
26         echo($exception->getMessage());
27         echo("</p>");
28         exit();
29     }
30
31     if($isConsult=="consultation"){
32         echo("<h1>Consultation Information:</h1>");
33
34         $info_cons = $connection->prepare(" SELECT *
35                                           FROM appointment
36                                           INNER JOIN consultation
37                                           USING (VAT_doctor,
38                                           date_timestamp)
39                                           INNER JOIN employee as
40                                           e
41                                           ON (e.VAT=c.VAT_doctor)
42                                           WHERE (c.VAT_doctor = :
43                                           VAT_doctor AND c.date_timestamp= :dates)");
44
45         $result=$info_cons->execute([
46             'VAT_doctor' => $VAT_doctor,
47             'dates' => $date_timestamp
48         ]);
49
50         $n_data = $info_cons->rowCount();
51         $data=$info_cons->fetchAll();
52
53         if($n_data>0)
54         {
55             echo("<table border=\"1\" style=\"margin-bottom: 20px
56             ;\"> ");
57             echo("<tr>");
58             echo("<td>VAT_client</td>");
59             echo("<td>Consultation Date and Time</td>");
60             echo("<td>Doctor's Name</td>");
61             echo("<td>Doctor's VAT</td>");
62             echo("<td>Appointment Description</td>");
63             echo("</tr>\n");
64             {
65                 foreach($data as $row){
66                     echo("<tr>");
67                     echo("<td>{$row['VAT_client']}'</td>");
68                     echo("<td>{$row['date_timestamp']}'</td>");
69                     echo("<td>{$row['name']}'</td>");
70                     echo("<td>{$row['VAT_doctor']}'</td>");
71                     echo("<td>{$row['description']}'</td>");
72                     echo("</tr>\n");
73                 }
74             }echo("</table>");

```

```

75         $info_ass = $connection->prepare(" SELECT *
76                                         FROM appointment
77                                         INNER JOIN
78                                         consultation as c
79                                         USING (VAT_doctor,
80                                         date_timestamp)
81                                         INNER JOIN (SELECT
82                                         *
83                                         FROM
84                                         consultation_assitant as ass
85                                         INNER
86                                         JOIN employee as e)
87                                         ON (e.
88                                         VAT=ass.VAT_nurse)
89                                         ) AS
90                                         nurse
91                                         USING (VAT_doctor,
92                                         date_timestamp)
93                                         WHERE (c.VAT_doctor
94                                         = :VAT_doctor AND c.date_timestamp= :dates)");
95
96     $result=$info_ass->execute([
97     'VAT_doctor' => $VAT_doctor,
98     'dates' => $date_timestamp
99     ]);
100    $n_ass = $info_ass->rowCount();
101    $ass=$info_ass->fetchAll();
102    if($n_ass>0){
103        echo("</table>");
104        echo("<table border=\"1\" style=\"margin-bottom: 20
105px;\">> ");
106
107        echo("<tr>");
108        echo("<h2>SOAP:</h2>");
109        echo("<td>Nurse name</td>");
110        echo("<td>Nurse VAT</td>");
111        {
112            foreach($ass as $row){
113                echo("<tr>");
114                echo("<td>{$row['VAT_nurse']}</td>");
115                echo("<td>{$row['name']}</td>");
116                echo("</tr>\n");
117            }
118        } echo("</table>");
119    }else{
120        echo("<td>No nurse present</td>");
121    }
122
123    echo("<table border=\"1\" style=\"margin-bottom: 20px
124;\">> ");
125
126    echo("<tr>");
127    echo("<h2>SOAP:</h2>");
128    echo("<td>Subjective Observation</td>");
129    echo("<td>Objective Observation</td>");
130    echo("<td>Assessment</td>");
131    echo("<td>Plan</td>");
132    echo("</tr>\n");
133    {

```

```

122         foreach($data as $row){
123             echo("<tr>");
124             echo("<td>{$row['s']}

```

```

168                                     FROM appointment as
169     a                                     INNER JOIN
170     consultation as c                     USING (VAT_doctor,
171     date_timestamp)                       INNER JOIN
172     prescription as pres                 USING(VAT_doctor,
173                                     WHERE (c.VAT_doctor
=:VAT_doctor AND c.date_timestamp= :dates));
174     $result=$info_pres->execute([
175     'VAT_doctor' => $VAT_doctor,
176     'dates' => $date_timestamp
177     ]);
178     $n_pres = $info_pres->rowCount();
179     $pres=$info_pres->fetchAll();
180     if($n_pres>0)
181     {
182         echo("<h2>Prescriptions:</h2>");
183         echo("</table>");
184         echo("<table border=\"1\" style=\"margin-bottom: 20
px;\> ");
185         echo("<tr>");
186         echo("<td>Prescription</td>");
187         echo("<td>Lab</td>");
188         echo("<td>Dosage</td>");
189         echo("<td>Description</td>");
190
191         echo("</tr>\n");
192         {
193             foreach($pres as $row){
194                 echo("<tr>");
195                 echo("<td>{$row['name']}</td>");
196                 echo("<td>{$row['lab']}</td>");
197                 echo("<td>{$row['dosage']}</td>");
198                 echo("<td>{$row['description']}</td>");
199                 echo("</tr>\n");
200             }
201             }echo("</table>");
202         }
203     }else{
204         echo("No more information found");
205     }
206 }else{
207     $info_apps = $connection->prepare("SELECT *
208                                     FROM appointment as a
209                                     INNER JOIN employee as
e
210                                     ON (e.VAT=a.VAT_doctor)
211                                     WHERE (a.VAT_doctor = :
VAT_doctor AND a.date_timestamp= :dates));
212     $result=$info_apps->execute(['VAT_doctor' => $VAT_doctor, '
dates' => $date_timestamp]);
213
214     $n_info = $info_apps->rowCount();
215     if($n_info>0)

```

```

216         {
217             echo("<h1>Appointment Information:</h1>");
218
219             echo("<table border=\"1\" style=\"margin-bottom: 20px
;\> ");
220             echo("<tr>");
221             echo("<td>Date and Time</td>");
222             echo("<td>Doctor's name</td>");
223             echo("<td>Doctor's VAT</td>");
224             echo("<td>Description</td>");
225             echo("</tr>\n");
226         {
227             while($apps=$info_apps->fetch()){
228                 echo("<tr>");
229                 echo("<td>{$apps['date_timestamp']}</td>");
230                 echo("<td>{$apps['name']}</td>");
231                 echo("<td>{$apps['VAT_doctor']}</td>");
232                 echo("<td>{$apps['description']}</td>");
233                 echo("</tr>\n");
234             }
235         }
236         echo("</table>");
237     }else{ #
-----
238         echo("No more information found");
239     }
240 }
241
242 if($isConsult!="consultation"){
243     echo("<form action='add_info_form.php' method='post'>
244     <input type='submit' value='Add Information' />
245     <input type='hidden' name='VAT_doctor' value='{ $VAT_doctor }'>
246     <input type='hidden' name='VAT_client' value='{ $VAT_client }'>
247     <input type='hidden' name='date_timestamp' value='{
$date_timestamp}'>
248     </form>");
249 }
250 #####
251
252 $search_procedure = $connection->prepare("SELECT name,
description
253                                     FROM
procedure_in_consultation
254                                     WHERE (VAT_doctor=:
VAT_doctor AND date_timestamp= :dates)");
255 $result5=$search_procedure->execute([
256     'VAT_doctor' => $VAT_doctor,
257     'dates' => $date_timestamp
258 ]);
259 $n_procedures = $search_procedure->rowCount();
260 $procedures=$search_procedure->fetchAll();
261 if($n_procedures>0)
262 {
263     echo("<h2>Procedures:</h2>");
264     echo("</table>");
265     echo("<table border=\"1\" style=\"margin-bottom: 20
px;\> ");

```



```

266         echo("<tr>");
267         echo("<td>Name</td>");
268         echo("<td>Description</td>");
269         echo("</tr>\n");
270     {
271         foreach($procedures as $row){
272             echo("<tr>");
273             echo("<td>{$row['name']}</td>");
274             echo("<td>{$row['description']}</td>");
275             echo("</tr>\n");
276         }
277     }echo("</table>");
278 }
279
280     $search_charting = $connection->prepare("SELECT name, quadrant,
nm, description, measure
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
FROM
WHERE (VAT=:
VAT_doctor AND date_timestamp= :dates)");
$result6=$search_charting->execute([
'VAT_doctor' => $VAT_doctor,
'dates' => $date_timestamp
]);
$n_charting = $search_charting->rowCount();
$charts=$search_charting->fetchAll();
if($n_charting>0)
{
    echo("<h2>Chartings:</h2>");
    echo("</table>");
    echo("<table border=\"1\" style=\"margin-bottom: 20
px;\n> ");
    echo("<tr>");
    echo("<td>Name</td>");
    echo("<td>Quadrant</td>");
    echo("<td>Number</td>");
    echo("<td>Description</td>");
    echo("<td>Measure</td>");
    echo("</tr>\n");
    {
        foreach($charts as $row){
            echo("<tr>");
            echo("<td>{$row['name']}</td>");
            echo("<td>{$row['quadrant']}</td>");
            echo("<td>{$row['nm']}</td>");
            echo("<td>{$row['description']}</td>");
            echo("<td>{$row['measure']}</td>");
            echo("</tr>\n");
        }
    }echo("</table>");
}
else
{
    echo("<form action='add_dental_form.php' method='
post'>
    <input type='submit' value='Add Dental Charting
Informations'>

```

```

317         <input type=hidden name='VAT_doctor' value='{
$VAT_doctor}'>
318         <input type=hidden name='date_timestamp' value='{
$date_timestamp}'>
319         <input type=hidden name='isConsult' value='{
$isConsult}'>
320         <input type=hidden name='VAT_client' value='{
$VAT_client}'>
321         </form>");
322     }
323
324     #####
325
326     echo("<form action='apps_and_cons.php' method='post'>
327     <input type=hidden name='VAT_client' value='{ $VAT_client}'>
328     <input type='submit' value='Go Back' /></a>
329     </form>");
330     # END CONNECTION
331     $connection = null;
332     ?>
333
334 </body>
335 </html>

```

6.5 Book Appointment

6.5.1 add_app_form.php

Choose Date:

Appointment Date:

Appointment Hour: Office hours are 9am to 5pm

Description:

Figura 31: Marcar Appointment

Listing 23: Codigo add_app_form.php

```

1 <html>
2 <body>
3
4 <h2>Choose Date:</h2>
5     <form action="search_doc.php" method="post">
6         <p>Appointment Date:
7             <input type="date" name="app_date" min='<?php echo date("Y-
m-d");?>' required/>

```

```

8      </p>
9      <p>Appointment Hour:
10         <input type="time" name="app_time" min="09:00:00" max="
11         17:00:00" step="3600" required/>
12         <small>Office hours are 9am to 5pm</small>
13     </p>
14     <p>
15         <input type=hidden name='VAT_client' value='<?php echo (
16         integer)$_REQUEST['VAT_client'];?>' required/>
17     </p>
18     <p>Description:
19         <input type=Text name='description' required/>
20     </p>
21     <p>
22         <input type="submit" value="Search For Doctor"/>
23     </p>
24     </form>
25     <a href="javascript:history.go(-1)">
26         <input type='submit' value='Go Back' /></a>
27 </form>
28 </body>
29 </html>

```

6.5.2 book_app.php

Appointment booked

Go Back

Figura 32: Appointment marcada

Listing 24: Código book_app.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5         $date = $_REQUEST['date'];
6         $time = $_REQUEST['time'];
7         $datetime_new = date('Y-m-d H:i:s', strtotime("$date $time"));
8
9         $host = "db.tecnico.ulisboa.pt";
10        $description= (string)$_REQUEST['description'];
11        $VAT_client= $_REQUEST['VAT_client'];
12        $VAT_doctor= $_REQUEST['VAT_doctor'];
13        # TRY TO CONNECT WITH DB
14
15        $host = "db.tecnico.ulisboa.pt";

```

```
16     $user = "ist425487";
17     $pass = "vief8835";
18     $dsn = "mysql:host=$host;dbname=$user";
19     $connection = null;
20     try
21     {
22         $connection = new PDO($dsn, $user, $pass);
23     }
24     catch(PDOException $exception)
25     {
26         echo("<p>Error: ");
27         echo($exception->getMessage());
28         echo("</p>");
29         exit();
30     }
31
32     $stmt=$connection->prepare("INSERT INTO appointment (VAT_doctor
, date_timestamp, description, VAT_client)
33                               VALUES (:VAT_doctor, :datetime_new,
:description, :VAT_client)");
34     $stmt->bindParam(':VAT_doctor', $VAT_doctor);
35     $stmt->bindParam(':datetime_new', $datetime_new);
36     $stmt->bindParam(':description', $description);
37     $stmt->bindParam(':VAT_client', $VAT_client);
38     $result=$stmt->execute();
39     if ($result == FALSE)
40     {
41         echo("Couldn't book appointment");
42     }else{
43         echo("Appointment booked");
44     }
45     echo ("<p></p>");
46
47     echo("<form action='apps_and_cons.php' method='post'>
48     <input type='hidden' value='{ $VAT_client}' name='VAT_client'>
49     <input type='submit' value='Go Back' />
50     </form>");
51
52     # END CONNECTION
53     $connection = null;
54     ?>
55 </body>
56 </html>
```

6.5.3 search_doc.php

date 2022-01-04

time 09:00:00

Number of Doctors available: 5

VAT_doctor	Name	
10001	Ada Lovelace	Book Appointment
10002	Jane Sweettooth	Book Appointment
10050	Jonh Sweettooth	Book Appointment
10051	Malcolm Middle	Book Appointment
10052	Ariel Red Sea	Book Appointment

Figura 33: Procura por um Doutor

Listing 25: Codigo search_doc.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5         $date = $_REQUEST['app_date'];
6         $time = $_REQUEST['app_time'];
7         $VAT_client= $_REQUEST['VAT_client'];
8         $description= $_REQUEST['description'];
9
10        # TRY TO CONNECT WITH DB
11        $host = "db.tecnico.ulisboa.pt";
12        $user = "ist425487";
13        $pass = "vief8835";
14        $dsn = "mysql:host=$host;dbname=$user";
15        $connection = null;
16        try
17        {
18            $connection = new PDO($dsn, $user, $pass);
19        }
20        catch(PDOException $exception)
21        {
22            echo("<p>Error: ");
23            echo($exception->getMessage());
24            echo("</p>");
25            exit();
26        }
27
28        if(empty($date) || empty($time))
29        {

```

```

30         echo("Appointment date or time not complete.");
31         echo ("<a href=\"javascript:history.go(-1)\">
32 <input type='submit' value='Go Back' /></a>
33 </form>");
34     }
35     else
36     {
37         echo("<p>date {$date}</p>");
38         $time= date('H:i:s', strtotime("$time"));
39         echo("<p>time {$time}</p>");
40         $find_doctors = $connection->prepare("SELECT *
41                                           FROM employee as e
42                                           WHERE e.VAT NOT IN (
43
44 SELECT VAT_doctor
45
46 FROM appointment
47
48 WHERE DATE(date_timestamp) =:appointment_date AND TIME(
49 date_timestamp) =:appointment_time)
50
51 AND e.VAT IN(SELECT VAT
52 FROM doctor)");
53
54 $result=$find_doctors ->execute([
55 'appointment_date' => $date,
56 'appointment_time' => $time,
57 ]);
58 if ($result== FALSE)
59 {
60     echo("Doctor search query.");
61     $info = $connection->errorInfo();
62     echo("<p>Error: {$info[1]}</p>");
63     exit();
64 }
65
66 $n_doctors=$find_doctors->rowCount();
67 if($n_doctors>0)
68 {
69     echo("<p> Number of Doctors available: {$n_doctors}</p>
70 " );
71     echo("<table border=\"1\" style=\"margin-bottom: 20px
72 ;\"> ");
73
74     echo("<tr>");
75     echo("<td>VAT_doctor</td>");
76     echo("<td>Name</td>");
77     echo("<td></td>");
78     echo("</tr>\n");
79
80     {
81         while($doctors=$find_doctors->fetch()){
82             echo("<tr>");
83             echo("<td>{$doctors['VAT']}'</td>");
84             echo("<td>{$doctors['name']}'</td>");
85
86             echo("<td><a href=\"book_app.php?VAT_client=");
87             echo("{$VAT_client}");
88             echo("&VAT_doctor=");
89             echo($doctors['VAT']);
90             echo("&date=");
91             echo($date);

```

```
81         echo("&time=");
82         echo($time);
83         echo("&description=");
84         echo($description);
85         echo("\>Book Appointment</a></td>\n");
86         echo("</tr>\n");
87     }
88 }
89 }echo("</table>");
90 }
91 #
92 #####
93
94     ###          PROCURAR MEDICOS PARA A DATA
95     ###
96     ###          SELECIONAR UM MEDICO PARA MARCAR COM ESSE
97     ###
98     #
99 #####
100
101     echo ("<a href=\"javascript:history.go(-1)\">
102     <input type='submit' value='Go Back' /></a>
103     </form>");
104     # END CONNECTION
105     $connection = null;
106     ?>
107
108 </body>
109 </html>
```

6.6 Add information to an Appointment

6.6.1 add_info_form.php

Information:

Please make sure to fill the SOAP criteria first:

SOAP:

Subjective Observation:

Objective Observation:

Assessment:

Plan:

Assistant Nurse(s) (Optional):

Nurse Name:

Diagnosis (Optional):

Diagnostic Code:

Prescription(s) (Optional):

Add a Diagnosis First

Figura 34: Adicionar informação a uma Appointment

Information:

Please make sure to fill the SOAP criteria first:

SOAP:

Subjective Observation:	Objective Observation:	Assessment:	Plan:
Bad teeth	muitas caries	nao sei	substituir os dentes todos

Assistant Nurse(s) (Optional):

Nurse's Name	Nurse's VAT
Buffy The Vampire Slayer	10202

Nurse Name:

Diagnosis (Optional):

Diagnostic Code
46557008

Diagnostic Code:

Prescription(s) (Optional):

Prescription ID	Prescription names	Prescription lab	Prescription dosage	Prescription description
46557008	ben-u-ron	Ibu Lab	2	per day for a week

Prescription for ID 46557008:

Name of medication: Dosage: Description:

Figura 35: Adicionar mais informação a uma Appointment

Listing 26: Codigo add_info_form.php

```

1 <html>
2 <body>
3     <h1>Information:</h1>
4     <h3>Please make sure to fill the SOAP criteria first:</h3>
5 <!-- ##### SOAP
6     ##### -->
7 <h2>SOAP:</h2>
8     <?php
9
10         $VAT_client = (integer)$_REQUEST['VAT_client'];
11         $VAT_doctor=(integer)$_REQUEST['VAT_doctor'];
12         $date_timestamp=(string)$_REQUEST['date_timestamp'];
13         # TRY TO CONNECT WITH DB
14         $host = "db.tecnico.ulisboa.pt";
15         $user = "ist425487";
16         $pass = "vief8835";
17         $dsn = "mysql:host=$host;dbname=$user";
18         $connection = null;
19         try
20         {
21             $connection = new PDO($dsn, $user, $pass);

```

```

22     }
23     catch(PDOException $exception)
24     {
25         echo("<p>Error: ");
26         echo($exception->getMessage());
27         echo("</p>");
28         exit();
29     }
30
31     // VER SE JA EXISTEM SOAP NOTES. SE SIM, PUBLICAR TABELA
32     $sql1= "SELECT s, o, a, p FROM consultation INNER JOIN
appointment USING (VAT_doctor, date_timestamp) WHERE (VAT_doctor=:
VAT_doctor AND date_timestamp=:date_timestamp)";
33     $result = $connection->prepare($sql1);
34     $result1=$result->execute([
35         'VAT_doctor' => $VAT_doctor,
36         'date_timestamp' => $date_timestamp
37     ]);
38
39     if ($result1 == FALSE)
40     {
41         $info = $connection->errorInfo();
42         echo("<p>Error: {$info[2]}</p>");
43     }
44     $n_data = $result->rowCount(); // number of nurses
45     $data=$result->fetchAll();
46     if($n_data>0){
47
48         $isConsult="consultation";
49         echo("<table border=\"1\" style=\"margin-bottom: 20px
;\"> ");
50
51         echo("<tr>");
52         echo("<td>Subjective Observation:</td>");
53         echo("<td>Objective Observation:</td>");
54         echo("<td>Assessment:</td>");
55         echo("<td>Plan:</td>");
56         echo("</tr>\n");
57     {
58         foreach($data as $row)
59         {
60             echo("<tr>");
61             echo("<td>{$row['s']}</td>");
62             echo("<td>{$row['o']}</td>");
63             echo("<td>{$row['a']}</td>");
64             echo("<td>{$row['p']}</td>");
65             echo("</tr>\n");
66         }
67         echo("</table>");
68     }else{
69         $today = date("Y-m-d H:i:s");
70         if ($date_timestamp < $today) {$isConsult="missed";}
71         else{$isConsult="booked";}
72
73         echo("<form action='add_soap.php' method='post'>");
74         echo("<p>Subjective Observation:");
75         echo("<input type='text' maxlength='255' name='sub'
required/>");

```

```

76         echo("</p>");
77         echo("<p>Objective Observation:");
78         echo("<input type='text' name='obj' maxlength='255'
required/>");
79         echo("</p>");
80         echo("<p>Assessment:");
81         echo("<input type='text' name='assess' maxlength='255'
required/>");
82         echo("</p>");
83         echo("<p>Plan:");
84         echo("<input type='text' name='plan' maxlength='255'
required/>");
85         echo("</p>");
86         echo("<input type=hidden name='VAT_client' value='{
$VAT_client}'>");
87         echo("<input type='hidden' value='{ $VAT_doctor}' name='
VAT_doctor' />");
88         echo("<input type='hidden' value='{ $date_timestamp}'
name='date_timestamp' />");
89         echo("<input type='submit' value='Add SOAP'>");
90         echo("</form>");
91     }
92     $connection=null;
93     ?>
94
95 <!-- ##### NURSE
##### -->
96     <h2>Assistant Nurse(s) (Optional):</h2>
97     <?php
98         $VAT_client = (integer)$_REQUEST['VAT_client'];
99         $VAT_doctor=(integer)$_REQUEST['VAT_doctor'];
100         $date_timestamp=(string)$_REQUEST['date_timestamp'];
101         # TRY TO CONNECT WITH DB
102         $host = "db.tecnico.ulisboa.pt";
103         $user = "ist425487";
104         $pass = "vief8835";
105         $dsn = "mysql:host=$host;dbname=$user";
106         $connection = null;
107         try
108         {
109             $connection = new PDO($dsn, $user, $pass);
110         }
111         catch(PDOException $exception)
112         {
113             echo("<p>Error: ");
114             echo($exception->getMessage());
115             echo("</p>");
116             exit();
117         }
118
119         // VER SE JA EXISTEM NURSES. SE SIM, PUBLICAR TABELA
120         $sql1= "SELECT DISTINCT name,VAT FROM employee INNER JOIN
nurse USING (VAT) WHERE VAT in (SELECT VAT_nurse FROM
consultation_assistant WHERE (VAT_doctor=:VAT_doctor AND
date_timestamp=:date_timestamp)) ORDER BY name ASC";
121         $result = $connection->prepare($sql1);
122         $result1=$result->execute([
123             'VAT_doctor' => $VAT_doctor,

```

```

124         'date_timestamp' => $date_timestamp
125     ]);
126
127     if ($result1 == FALSE)
128     {
129         $info = $connection->errorInfo();
130         echo("<p>Error: {$info[2]}</p>");
131
132     }
133
134     $n_data = $result->rowCount(); // number of nurses
135     $data=$result->fetchAll();
136     if($n_data>0){
137         echo("<table border='\"1\"' style='\"margin-bottom: 20px";
        ;\ ">" );
            echo("<tr>");
            echo("<td>Nurse's Name</td>");
            echo("<td>Nurse's VAT</td>");
            echo("</tr>\n");
            {
                foreach($data as $row)
                {
                    echo("<tr>");
                    echo("<td>{$row['name']}</td>");
                    echo("<td>{$row['VAT']}</td>");
                    echo("</tr>\n");
                }
            }
            echo("</table>");
        }
        // SELECIONAR MAIS NURSES


138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
    echo("<form action='add_nurse.php' method='post'>");
    echo("<p>Nurse Name:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~");
    echo("<select name='name_nurse'>");
        $sql2= "SELECT DISTINCT name FROM employee INNER JOIN
nurse USING (VAT) WHERE VAT not in (SELECT VAT_nurse FROM
consultation_assistant WHERE (VAT_doctor=:VAT_doctor AND
date_timestamp=:date_timestamp)) ORDER BY name ASC";
        $result = $connection->prepare($sql2);
        $result1=$result->execute([
            'VAT_doctor' => $VAT_doctor,
            'date_timestamp' => $date_timestamp
        ]);
        if ($result1 == FALSE)
        {
            $info = $connection->errorInfo();
            echo("<p>Error: {$info[2]}</p>");
            exit();
        }
        $data=$result->fetchAll();
        foreach($data as $row)
        {
            $name = $row['name'];
            echo("<option value='$name'$>$name</option>");
        }
        echo("</select>&nbsp;&nbsp;&nbsp;&nbsp;&~");

```

```

178         echo("<input type=hidden name='VAT_client' value='{
$VAT_client}'>");
179         echo("<input type='hidden' value='{ $VAT_doctor}' name='
VAT_doctor' />");
180         echo("<input type='hidden' value='{ $date_timestamp}' name='
date_timestamp' />");
181         $n_data = $result->rowCount(); // number of nurses
182         if($n_data>0){
183             echo("<input type='submit' value='Add nurse'>");
184         }
185         echo("</form>");
186         echo("</p>");
187         $connection=null;
188     ?>
189
190
191 <!-- ##### DIAGNOSIS
##### -->
192
193
194 <h2>Diagnosis (Optional):</h2>
195 <?php
196     $VAT_client = (integer)$_REQUEST['VAT_client'];
197     $VAT_doctor=(integer)$_REQUEST['VAT_doctor'];
198     $date_timestamp=(string)$_REQUEST['date_timestamp'];
199     # TRY TO CONNECT WITH DB
200     $host = "db.tecnico.ulisboa.pt";
201     $user = "ist425487";
202     $pass = "vief8835";
203     $dsn = "mysql:host=$host;dbname=$user";
204     $connection = null;
205     try
206     {
207         $connection = new PDO($dsn, $user, $pass);
208     }
209     catch(PDOException $exception)
210     {
211         echo("<p>Error: ");
212         echo($exception->getMessage());
213         echo("</p>");
214         exit();
215     }
216     $sql1="SELECT DISTINCT ID FROM consultation_diagnostic
WHERE (VAT_doctor=:VAT_doctor AND date_timestamp=:date_timestamp)
ORDER BY ID ASC";
217     $result = $connection->prepare($sql1);
218     $result1=$result->execute([
219         'VAT_doctor' => $VAT_doctor,
220         'date_timestamp' => $date_timestamp
221     ]);
222     if ($result1 == FALSE)
223     {
224         $info = $connection->errorInfo();
225         echo("<p>Error: {$info[2]}</p>");
226     }
227     $n_data = $result->rowCount(); // number of nurses
228     $data=$result->fetchAll();
229     if($n_data>0){

```

[illegible]

```

279 <!-- ##### PRESCRIPTIONS
      ##### -->
280
281 <h2>Prescription(s) (Optional):</h2>
282 <?php
283
284     $VAT_client = (integer)$_REQUEST['VAT_client'];
285     $VAT_doctor=(integer)$_REQUEST['VAT_doctor'];
286     $date_timestamp=(string)$_REQUEST['date_timestamp'];
287
288     # TRY TO CONNECT WITH DB
289     $host = "db.tecnico.ulisboa.pt";
290     $user = "ist425487";
291     $pass = "vief8835";
292     $dsn = "mysql:host=$host;dbname=$user";
293     $connection = null;
294     try
295     {
296         $connection = new PDO($dsn, $user, $pass);
297     }
298     catch(PDOException $exception)
299     {
300         echo("<p>Error: ");
301         echo($exception->getMessage());
302         echo("</p>");
303         exit();
304     }
305
306
307     $sql1="SELECT DISTINCT ID, name, lab ,dosage, description
FROM prescription
308         WHERE (VAT_doctor=:VAT_doctor AND date_timestamp=:
date_timestamp) ORDER BY ID ASC ";
309     $result = $connection->prepare($sql1);
310     $result1=$result->execute([
311         'VAT_doctor' => $VAT_doctor,
312         'date_timestamp' => $date_timestamp
313     ]);
314     if ($result1 == FALSE)
315     {
316         $info = $connection->errorInfo();
317         echo("<p>Error: {$info[2]}</p>");
318     }
319     $n_data = $result->rowCount(); // number of prescriptions
320     $data=$result->fetchAll();
321     if($n_data>0){
322         echo("<table border=\"1\" style=\"margin-bottom: 20px
;\">");
323
324         echo("<tr>");
325         echo("<td>Prescription ID</td>");
326         echo("<td>Prescription names</td>");
327         echo("<td>Prescription lab</td>");
328         echo("<td>Prescription dosage</td>");
329         echo("<td>Prescription description</td>");
330         echo("</tr>\n");
331         {
332             foreach($data as $row)

```

[illegible]


```

387         $lab = $row['lab'];
388         echo("<option value=\"{$name}\">{$name}</option>");
389     }
390     echo("</select>&nbsp;&nbsp;&nbsp;&nbsp;");
391     echo("Dosage:");
392     echo("<input type='text' name='dosage' maxlength
='255' />");
393     echo("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;");
394     echo("Description:");
395     echo("<input type='text' name='description'
maxlength='255' />");
396     echo("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;");
397     if($n_data>0){
398         echo("<input type='submit' value='Add
Prescription'>");
399     }
400 }
401 echo("<input type='hidden' value='{$ID}' name='ID' />");
402 ;
403     echo("<input type=hidden name='VAT_client' value='{$
$VAT_client}'>");
404     echo("<input type='hidden' value='{$VAT_doctor}' name='
VAT_doctor' />");
405     echo("<input type='hidden' value='{$date_timestamp}'
name='date_timestamp' />");
406
407     echo("</form>");
408     echo("</p>");
409 }
410 }
411 else
412 {
413     echo("Add a Diagnosis First");
414 }
415 // SELECIONAR MAIS PRESCRIPTIONS
416
417 $connection=null;
418 echo("<p></p>");
419 echo("<form action='list_cons_info.php' method='post'>
420     <input type='hidden' value='{$VAT_client}' name='VAT_client
421     <input type='hidden' value='{$isConsult}' name='isConsult'>
422     <input type='hidden' value='{$VAT_doctor}' name='VAT_doctor
423     <input type='hidden' value='{$date_timestamp}' name='
date_timestamp' />
424     <input type='submit' value='Done' />
425     </form>");
426     ?>
427
428 <p></p>
429 <form action="initial_page.php" method="post">
430 <input type="submit" value="Go to Homepage" />
431 </form>
432 </body>
433 </html>

```

6.6.2 add_soap.php

Listing 27: Codigo add_soap.php

```

1  <html>
2  <body>
3      <?php
4          $VAT_client = (integer)$_REQUEST['VAT_client'];
5          $VAT_doctor= (integer)$_REQUEST['VAT_doctor'];
6          $date_timestamp= $_REQUEST['date_timestamp'];
7          $S = (string)$_REQUEST['sub'];
8          $O = (string)$_REQUEST['obj'];
9          $A = (string)$_REQUEST['assess'];
10         $P = (string)$_REQUEST['plan'];
11
12         # TRY TO CONNECT WITH DB
13         $host = "db.tecnico.ulisboa.pt";
14         $user = "ist425487";
15         $pass = "vief8835";
16         $dsn = "mysql:host=$host;dbname=$user";
17         $connection = null;
18         try
19         {
20             $connection = new PDO($dsn, $user, $pass);
21         }
22         catch(PDOException $exception)
23         {
24             echo("<p>Error: ");
25             echo($exception->getMessage());
26             echo("</p>");
27             exit();
28         }
29
30         #
31         #####
32
33         ###          ADICIONAR NOTAS a CONSULTA
34         ###
35         #
36         #####
37
38         $sql1 = "INSERT INTO consultation VALUES ($VAT_doctor, '
39         $date_timestamp', '$S','$O','$A','$P')";
40
41         $nrows = $connection->exec($sql1);
42         if($nrows>0)
43         {
44             echo("<p>SOAP notes included!</p>");
45         }
46         elseif ($nrows== FALSE)
47         {
48             echo("<p>Couldn't include SOAP notes</p>");
49             $info = $connection->errorInfo();
50             echo("<p>Error : {$info[1]}</p>");
51             exit();
52         }
53     }
54 }

```

```

49     header('Location: http://web.tecnico.ulisboa.pt/ist425487/
add_info_form.php?VAT_doctor='.$VAT_doctor.'&date_timestamp='.
$date_timestamp.'&VAT_client='.$VAT_client);
50
51     die();
52     $connection = null;
53     ?>
54
55 </body>
56 </html>

```

6.6.3 add_nurse.php

Listing 28: Codigo add_nurse.php

```

1 <html>
2 <body>
3     <?php
4         $VAT_client = (integer)$_REQUEST['VAT_client'];
5         $name_nurse=(string)$_REQUEST['name_nurse'];
6         $VAT_doctor= (integer)$_REQUEST['VAT_doctor'];
7         $date_timestamp= $_REQUEST['date_timestamp'];
8
9         # TRY TO CONNECT WITH DB
10        $host = "db.tecnico.ulisboa.pt";
11        $user = "ist425487";
12        $pass = "vief8835";
13        $dsn = "mysql:host=$host;dbname=$user";
14        $connection = null;
15        try
16        {
17            $connection = new PDO($dsn, $user, $pass);
18        }
19        catch(PDOException $exception)
20        {
21            echo("<p>Error: ");
22            echo($exception->getMessage());
23            echo("</p>");
24            exit();
25        }
26
27
28
29        ##### FIND CONSULTATION #####
30        $find_cons = $connection->prepare("SELECT *
31                                           FROM consultation
32                                           WHERE VAT_doctor=:
VAT_doctor AND date_timestamp=:date_timestamp");
33
34        $result3=$find_cons ->execute([
35            'VAT_doctor' => $VAT_doctor,
36            'date_timestamp'=> $date_timestamp,
37        ]);
38        if ($result3==FALSE)
39        {
40            echo("Couldn't find consultation.");

```

```

41         $info = $connection->errorInfo();
42         echo("<p>Error: {$info[1]}</p>");
43         exit();
44     }
45     $cons = $find_cons->rowCount();
46     if ($cons == 0)
47     {
48         echo("<p>There isn't a consultation associated to this
appointment.</p><p> Please create one first by adding SOAP
information to the appointment.</p>");
49         echo("<form action='add_info_form.php' method='post'>
50             <input type='hidden' name='VAT_client' value='{$VAT_client}'>
51             <input type='hidden' name='VAT_doctor' value='{$VAT_doctor}'>
52             <input type='hidden' name='date_timestamp' value='{$
$date_timestamp}'>
53             <input type='submit' value='Go Back' /></a>
54             </form>");
55     }
56     else
57     {
58         ##### NURSE
59         #####
60         $find_nurse_vat = $connection->prepare("SELECT VAT
61                                                     FROM employee
62                                                     WHERE name=:
name_nurse");
63
64         $result=$find_nurse_vat ->execute([
65             'name_nurse' => $name_nurse,
66         ]);
67         if ($result== FALSE)
68         {
69             echo("Couldn't find nurse");
70             $info = $connection->errorInfo();
71             echo("<p>Error: {$info[1]}</p>");
72             exit();
73         }
74         $data=$find_nurse_vat->fetchAll();
75         foreach($data as $row){
76             $VAT_nurse=$row['VAT'];
77         }
78
79         $sql2 = "INSERT INTO consultation_assistant VALUES (
$VAT_doctor, '$date_timestamp', $VAT_nurse)";
80         $nrows = $connection->exec($sql2);
81         if($nrows== FALSE)
82         {
83             echo("<p>Couldn't add nurse info</p>");
84             $info = $connection->errorInfo();
85             echo("<p>Error: {$info[1]}</p>");
86             exit();
87         }
88
89         header('Location: http://web.tecnico.ulisboa.pt/ist425487/
add_info_form.php?VAT_doctor='.$VAT_doctor.'&date_timestamp='.
$date_timestamp.'&VAT_client='.$VAT_client);
90         die();

```

```

91     }
92     $connection = null;
93     ?>
94
95     </body>
96     </html>

```

6.6.4 add_diagnostic.php

Listing 29: Código add_diagnostic.php

```

1 <html>
2 <body>
3     <?php
4         $VAT_client = (integer)$_REQUEST['VAT_client'];
5         $VAT_doctor= (integer)$_REQUEST['VAT_doctor'];
6         $date_timestamp= $_REQUEST['date_timestamp'];
7         $ID = (integer)$_REQUEST['ID'];
8
9
10        # TRY TO CONNECT WITH DB
11        $host = "db.tecnico.ulisboa.pt";
12        $user = "ist425487";
13        $pass = "vief8835";
14        $dsn = "mysql:host=$host;dbname=$user";
15        $connection = null;
16        try
17        {
18            $connection = new PDO($dsn, $user, $pass);
19        }
20        catch(PDOException $exception)
21        {
22            echo("<p>Error: ");
23            echo($exception->getMessage());
24            echo("</p>");
25            exit();
26        }
27
28        ##### FIND CONSULTATION #####
29        $find_cons = $connection->prepare("SELECT *
30                                           FROM consultation
31                                           WHERE VAT_doctor=:
32VAT_doctor AND date_timestamp=:date_timestamp");
33
34        $result3=$find_cons ->execute([
35            'VAT_doctor' => $VAT_doctor,
36            'date_timestamp'=> $date_timestamp,
37        ]);
38        if ($result3==FALSE)
39        {
40            echo("Couldn't find consultation.");
41            $info = $connection->errorInfo();
42            echo("<p>Error: {$info[1]}</p>");
43            exit();
44        }
45        $cons = $find_cons->rowCount();

```

```

45     if ($cons == 0)
46     {
47         echo("<p>There isn't a consultation associated to this
appointment.</p><p> Please create one first by adding SOAP
information to the appointment.</p>");
48         echo("<form action='add_info_form.php' method='post'>
49             <input type='hidden' name='VAT_client' value='{ $VAT_client }'>
50             <input type='hidden' name='VAT_doctor' value='{ $VAT_doctor }'>
51             <input type='hidden' name='date_timestamp' value='{
$date_timestamp }'>
52             <input type='submit' value='Go Back' /></a>
53             </form>");
54     }
55     else
56     {
57         $stmt=$connection->prepare("INSERT INTO
consultation_diagnostic
58                                     VALUES ( $VAT_doctor , '
$date_timestamp', $ID)");
59         $result=$stmt->execute();
60         if ($result == FALSE)
61         {
62             echo("<p>Couldn't add Diagnostic in consultation</p>");
63             $info = $connection->errorInfo();
64             echo("<p>Error: { $info[1] }</p>");
65             exit();
66         }
67
68         header('Location: http://web.tecnico.ulisboa.pt/ist425487/
add_info_form.php?VAT_doctor='.$VAT_doctor.'&date_timestamp='.
$date_timestamp.'&VAT_client='.$VAT_client);
69         die();
70     }
71     $connection = null;
72     ?>
73
74 </body>
75 </html>

```

6.6.5 add_prescription.php

Listing 30: Codigo add_prescription.php

```

1 <html>
2     <body>
3         <?php
4             $VAT_client = (integer)$_REQUEST['VAT_client'];
5             $name=(string)$_REQUEST['name'];
6             $ID=(integer)$_REQUEST['ID'];
7             $dosage=(string)$_REQUEST['dosage'];
8             $description=(string)$_REQUEST['description'];
9             $VAT_doctor= (integer)$_REQUEST['VAT_doctor'];
10            $date_timestamp= $_REQUEST['date_timestamp'];
11
12            # TRY TO CONNECT WITH DB
13            $host = "db.tecnico.ulisboa.pt";

```

```

14     $user = "ist425487";
15     $pass = "vief8835";
16     $dsn = "mysql:host=$host;dbname=$user";
17     $connection = null;
18     try
19     {
20         $connection = new PDO($dsn, $user, $pass);
21     }
22     catch(PDOException $exception)
23     {
24         echo("<p>Error: ");
25         echo($exception->getMessage());
26         echo("</p>");
27         exit();
28     }
29
30     ##### PRESCRIPTIONS
31     #####
32
33     $sql_aux="SELECT lab FROM medication WHERE (name=:name)";
34     $result = $connection->prepare($sql_aux);
35     $result1=$result->execute([
36         'name' => $name
37     ]);
38     if ($result1 == FALSE)
39     {
40         $info = $connection->errorInfo();
41         echo("<p>Error1: {$info[2]}</p>");
42         exit();
43     }
44     $n_data = $result->rowCount(); // number of medication
45     $data=$result->fetchAll();
46     if($n_data>0){
47         foreach($data as $row)
48         {
49             $lab = $row['lab'];
50         }
51     }
52
53
54     $sql_test="SELECT * FROM prescription WHERE (name=:name and
55     lab=:lab and VAT_doctor=:VAT_doctor and date_timestamp=:
56     date_timestamp and ID=:ID)";
57     $result = $connection->prepare($sql_test);
58     $result1=$result->execute([
59         'name' => $name,
60         'lab' => $lab,
61         'VAT_doctor' => $VAT_doctor,
62         'date_timestamp' => $date_timestamp,
63         'ID' => $ID
64     ]);
65     if ($result1 == FALSE)
66     {
67         $info = $connection->errorInfo();
68         echo("<p>Error2: {$info[1]}</p>");
69         exit();
70     }

```

```
69         $n_data = $result->rowCount();
70         echo("<p>n_data={$n_data}</p>");
71         if($n_data==0){
72             $sql="INSERT INTO prescription
73                 VALUES ('$name', '$lab', $VAT_doctor, '
74 $date_timestamp', $ID, '$dosage', '$description')";
75             $result = $connection->exec($sql);
76             if($result>0)
77             {
78                 echo("<p>Prescription included!</p>");
79             }
80             elseif ($result== FALSE)
81             {
82                 echo("<p>Couldn't include prescriptons</p>");
83                 $info = $connection->errorInfo();
84                 echo("<p>Error3 : {$info[1]}</p>");
85                 exit();
86             }
87             header('Location: http://web.tecnico.ulisboa.pt/ist425487
88 /add_info_form.php?VAT_doctor='.$VAT_doctor.'&date_timestamp='.
89 $date_timestamp.'&VAT_client='.$VAT_client);
90             die();
91         }
92         $connection = null;
93     ?>
94 </body>
95 </html>
```


6.7 Insert Dental Charting

6.7.1 add_dental_form.php

Add Dental Charting Information:

You must add a measurement and description for each teeth you want to add to the database.

Description:

Top Right:

Central Incisor:

Measurement: Description:

Lateral Incisor:

Measurement: Description:

Cuspid:

Measurement: Description:

First Bicuspид:

Measurement: Description:

Second Bicuspид:

Measurement: Description:

First Molar:

Measurement: Description:

Second Molar:

Measurement: Description:

Third Molar:

Figura 36: Adicionar Informação Dental Charting

Listing 31: Código add_dental_form.php

```

1 <html>
2 <body><h2>Add Dental Charting Information:</h2>
3 <h3>You must add a measurement and description for each teeth you
  want to add to the database.</h3>
4 <form action="add_dental.php" method="post">
5
6 <h3>Description:</h3>
7 <p>
8 <input type='text' maxlength='255' name='desc' required/>
9 </p>
10 <h3>Top Right:</h3>
11 <h4>Central Incisor:</h4>
12 <p>
13 Measurement:
14 <input type="number" min="0" name="TR1" />

```

```

15         Description:
16         <input type='text' maxlength='255' name='TR1D' />
17     </p>
18     <h4>Lateral Incisor:</h4>
19     <p>
20         Measurement:
21         <input type="number" min="0" name="TR2" />
22         Description:
23         <input type='text' maxlength='255' name='TR2D' />
24     </p>
25     <h4>Cuspid:</h4>
26     <p>
27         Measurement:
28         <input type="number" min="0" name="TR3" />
29         Description:
30         <input type='text' maxlength='255' name='TR3D' />
31     </p>
32     <h4>First Bicuspid:</h4>
33     <p>
34         Measurement:
35         <input type="number" min="0" name="TR4" />
36         Description:
37         <input type='text' maxlength='255' name='TR4D' />
38     </p>
39     <h4>Second Bicuspid:</h4>
40     <p>
41         Measurement:
42         <input type="number" min="0" name="TR5" />
43         Description:
44         <input type='text' maxlength='255' name='TR5D' />
45     </p>
46     <p>First Molar:</h4>
47     <p>
48         Measurement:
49         <input type="number" min="0" name="TR6" />
50         Description:
51         <input type='text' maxlength='255' name='TR6D' />
52     </p>
53     <h4>Second Molar:</h4>
54     <p>
55         Measurement:
56         <input type="number" min="0" name="TR7" />
57         Description:
58         <input type='text' maxlength='255' name='TR7D' />
59     </p>
60     <h4>Third Molar:</h4>
61     <p>
62         Measurement:
63         <input type="number" min="0" name="TR8" />
64         Description:
65         <input type='text' maxlength='255' name='TR8D' />
66     </p>
67
68     <h3>Top Left:</h3>
69     <h4>Central Incisor:</h4>
70     <p>
71         Measurement:
72

```

```

73         <input type="number" min="0" name="TL1" />
74         Description:
75         <input type='text' maxlength='255' name='TL1D' />
76     </p>
77     <h4>Lateral Incisor:</h4>
78     <p>
79         Measurement:
80         <input type="number" min="0" name="TL2" />
81         Description:
82         <input type='text' maxlength='255' name='TL2D' />
83     </p>
84     <h4>Cuspid:</h4>
85     <p>
86         Measurement:
87         <input type="number" min="0" name="TL3" />
88         Description:
89         <input type='text' maxlength='255' name='TL3D' />
90     </p>
91     <h4>First Bicuspid:</h4>
92     <p>
93         Measurement:
94         <input type="number" min="0" name="TL4" />
95         Description:
96         <input type='text' maxlength='255' name='TL4D' />
97     </p>
98     <h4>Second Bicuspid:</h4>
99     <p>
100         Measurement:
101         <input type="number" min="0" name="TL5" />
102         Description:
103         <input type='text' maxlength='255' name='TL5D' />
104     </p>
105     <h4>First Molar:</h4>
106     <p>
107         Measurement:
108         <input type="number" min="0" name="TL6" />
109         Description:
110         <input type='text' maxlength='255' name='TL6D' />
111     </p>
112     <h4>Second Molar:</h4>
113     <p>
114         Measurement:
115         <input type="number" min="0" name="TL7" />
116         Description:
117         <input type='text' maxlength='255' name='TL7D' />
118     </p>
119     <h4>Third Molar:</h4>
120     <p>
121         Measurement:
122         <input type="number" min="0" name="TL8" />
123         Description:
124         <input type='text' maxlength='255' name='TL8D' />
125     </p>
126
127     <h3>Bottom Right:</h3>
128     <h4>Central Incisor:</h4>
129     <p>
130         Measurement:

```

```

131         <input type="number" min="0" name="BR1" />
132         Description:
133         <input type='text' maxlength='255' name='BR1D' />
134     </p>
135     <h4>Lateral Incisor:</h4>
136     <p>
137         Measurement:
138         <input type="number" min="0" name="BR2" />
139         Description:
140         <input type='text' maxlength='255' name='BR2D' />
141     </p>
142     <h4>Cuspid:</h4>
143     <p>
144         Measurement:
145         <input type="number" min="0" name="BR3" />
146         Description:
147         <input type='text' maxlength='255' name='BR3D' />
148     </p>
149     <h4>First Bicuspid:</h4>
150     <p>
151         Measurement:
152         <input type="number" min="0" name="BR4" />
153         Description:
154         <input type='text' maxlength='255' name='BR4D' />
155     </p>
156     <h4>Second Bicuspid:</h4>
157     <p>
158         Measurement:
159         <input type="number" min="0" name="BR5" />
160         Description:
161         <input type='text' maxlength='255' name='BR5D' />
162     </p>
163     <h4>First Molar:</h4>
164     <p>
165         Measurement:
166         <input type="number" min="0" name="BR6" />
167         Description:
168         <input type='text' maxlength='255' name='BR6D' />
169     </p>
170     <h4>Second Molar:</h4>
171     <p>
172         Measurement:
173         <input type="number" min="0" name="BR7" />
174         Description:
175         <input type='text' maxlength='255' name='BR7D' />
176     </p>
177     <h4>Third Molar:</h4>
178     <p>
179         Measurement:
180         <input type="number" min="0" name="BR8" />
181         Description:
182         <input type='text' maxlength='255' name='BR8D' />
183     </p>
184
185     <h3>Bottom Left:</h3>
186     <h4>Central Incisor:</h4>
187     <p>
188         Measurement:

```

```

189         <input type="number" min="0" name="BL1" />
190         Description:
191         <input type='text' maxlength='255' name='BL1D' />
192     </p>
193     <h4>Lateral Incisor:</h4>
194     <p>
195         Measurement:
196         <input type="number" min="0" name="BL2" />
197         Description:
198         <input type='text' maxlength='255' name='BL2D' />
199     </p>
200     <h4>Cuspid:</h4>
201     <p>
202         Measurement:
203         <input type="number" min="0" name="BL3" />
204         Description:
205         <input type='text' maxlength='255' name='BL3D' />
206     </p>
207     <h4>First Bicuspid:</h4>
208     <p>
209         Measurement:
210         <input type="number" min="0" name="BL4" />
211         Description:
212         <input type='text' maxlength='255' name='BL4D' />
213     </p>
214     <h4>Second Bicuspid:</h4>
215     <p>
216         Measurement:
217         <input type="number" min="0" name="BL5" />
218         Description:
219         <input type='text' maxlength='255' name='BL5D' />
220     </p>
221     <h4>First Molar:</h4>
222     <p>
223         Measurement:
224         <input type="number" min="0" name="BL6" />
225         Description:
226         <input type='text' maxlength='255' name='BL6D' />
227     </p>
228     <h4>Second Molar:</h4>
229     <p>
230         Measurement:
231         <input type="number" min="0" name="BL7" />
232         Description:
233         <input type='text' maxlength='255' name='BL7D' />
234     </p>
235     <h4>Third Molar:</h4>
236     <p>
237         Measurement:
238         <input type="number" min="0" name="BL8" />
239         Description:
240         <input type='text' maxlength='255' name='BL8D' />
241     </p>
242     <p><input type="submit" value="Add"/></p>
243     <input type=hidden name='VAT_doctor' value='<?php echo (integer
244     $_REQUEST['VAT_doctor']);?>' required/>
245     <input type=hidden name='date_timestamp' value='<?php echo
246     $_REQUEST['date_timestamp'];?>' required/>

```

```

245     <input type=hidden name='isConsult' value='<?php echo $_REQUEST
    ['isConsult'];?>' required/>
246
247     </form>
248     <form action="javascript:history.go(-1)" method='post'>
249     <input type='submit' value='Go Back' />
250     </form>
251 </body>
252 </html>

```

6.7.2 add_dental.php

Listing 32: Codigo add_dental.php

```

1 <html>
2 <body>
3     <?php
4         # FORM
5
6         $TR = array($_REQUEST['TR1'], $_REQUEST['TR2'], $_REQUEST['TR3',
    ], $_REQUEST['TR4'], $_REQUEST['TR5'], $_REQUEST['TR6'], $_REQUEST['
    TR7'], $_REQUEST['TR8']);
7
8         $TL = array($_REQUEST['TL1'], $_REQUEST['TL2'], $_REQUEST['TL3',
    ], $_REQUEST['TL4'], $_REQUEST['TL5'], $_REQUEST['TL6'], $_REQUEST['
    TL7'], $_REQUEST['TL8']);
9
10        $BR = array($_REQUEST['BR1'], $_REQUEST['BR2'], $_REQUEST['BR3',
    ], $_REQUEST['BR4'], $_REQUEST['BR5'], $_REQUEST['BR6'], $_REQUEST['
    BR7'], $_REQUEST['BR8']);
11
12        $BL = array($_REQUEST['BL1'], $_REQUEST['BL2'], $_REQUEST['BL3',
    ], $_REQUEST['BL4'], $_REQUEST['BL5'], $_REQUEST['BL6'], $_REQUEST['
    BL7'], $_REQUEST['BL8']);
13
14
15
16        $TRD = array($_REQUEST['TR1D'], $_REQUEST['TR2D'], $_REQUEST['
    TR3D'], $_REQUEST['TR4D'], $_REQUEST['TR5D'], $_REQUEST['TR6D'],
    $_REQUEST['TR7D'], $_REQUEST['TR8D']);
17
18        $TLD = array($_REQUEST['TL1D'], $_REQUEST['TL2D'], $_REQUEST['
    TL3D'], $_REQUEST['TL4D'], $_REQUEST['TL5D'], $_REQUEST['TL6D'],
    $_REQUEST['TL7D'], $_REQUEST['TL8D']);
19
20        $BRD = array($_REQUEST['BR1D'], $_REQUEST['BR2D'], $_REQUEST['
    BR3D'], $_REQUEST['BR4D'], $_REQUEST['BR5D'], $_REQUEST['BR6D'],
    $_REQUEST['BR7D'], $_REQUEST['BR8D']);
21
22        $BLD = array($_REQUEST['BL1D'], $_REQUEST['BL2D'], $_REQUEST['
    BL3D'], $_REQUEST['BL4D'], $_REQUEST['BL5D'], $_REQUEST['BL6D'],
    $_REQUEST['BL7D'], $_REQUEST['BL8D']);
23
24        $description=$_REQUEST['desc'];
25
26        $VAT_doctor=(integer)$_REQUEST['VAT_doctor'];

```

```

27     $date_timestamp=(string)$_REQUEST['date_timestamp'];
28     $isConsult=$_REQUEST['isConsult'];
29
30
31     # TRY TO CONNECT WITH DB
32     $host = "db.tecnico.ulisboa.pt";
33     $user = "ist425487";
34     $pass = "vief8835";
35     $dsn = "mysql:host=$host;dbname=$user";
36     $connection = null;
37     try
38     {
39         $connection = new PDO($dsn, $user, $pass);
40     }
41     catch(PDOException $exception)
42     {
43         echo("<p>Error: ");
44         echo($exception->getMessage());
45         echo("</p>");
46         exit();
47     }
48
49     ##### FIND CLIENT VAT #####
50
51     $find_client = $connection->prepare("SELECT VAT_client
52                                         FROM appointment
53                                         WHERE VAT_doctor=:
54 VAT_doctor AND date_timestamp=:date_timestamp");
55
56     $result=$find_client ->execute([
57 'VAT_doctor' => $VAT_doctor,
58 'date_timestamp'=> $date_timestamp,
59 ]);
60 if ($result== FALSE)
61 {
62     echo("Couldn't find client.");
63     $info = $connection->errorInfo();
64     echo("<p>Error: {$info[1]}</p>");
65     exit();
66 }
67
68 $data=$find_client->fetchAll();
69 foreach($data as $row)
70 {
71     $VAT_client=$row['VAT_client'];
72 }
73
74     ##### FIND CLIENT AGE
75     #####
76     ## (to know which type of procedure to insert (teenager or
77     adult)) ##
78
79     $find_client_age = $connection->prepare("SELECT age
80                                         FROM client
81                                         WHERE VAT_client=:
82 VAT_client");

```

```

81     $result2=$find_client_age->execute([
82     'VAT_client' => $VAT_client,
83     ]);
84     if ($result2== FALSE)
85     {
86         echo("<p>Couldn't find client age.");
87         $info = $connection->errorInfo();
88         echo("<p>Error: {$info[1]}</p>");
89         exit();
90     }
91     $data=$find_client_age->fetchAll();
92     foreach($data as $row)
93     {
94         $client_age=$row['age'];
95     }
96
97     ##### FIND CONSULTATION #####
98     $find_cons = $connection->prepare("SELECT *
99                                     FROM consultation
100                                    WHERE VAT_doctor=:
VAT_doctor AND date_timestamp=:date_timestamp");
101
102     $result3=$find_cons ->execute([
103     'VAT_doctor' => $VAT_doctor,
104     'date_timestamp'=> $date_timestamp,
105     ]);
106     if ($result3==FALSE)
107     {
108         echo("Couldn't find consultation.");
109         $info = $connection->errorInfo();
110         echo("<p>Error: {$info[1]}</p>");
111         exit();
112     }
113     $cons = $find_cons->rowCount();
114     if ($cons == 0)
115     {
116         echo("<p>There isn't a consultation associated to this
appointment.</p><p> Please create one first by adding SOAP
information to the appointment.</p>");
117         echo("<form action='list_cons_info.php' method='post'>
118             <input type=hidden name='VAT_client' value='{ $VAT_client }'>
119             <input type=hidden name='VAT_doctor' value='{ $VAT_doctor }'>
120             <input type=hidden name='date_timestamp' value='{
$date_timestamp }'>
121             <input type=hidden name='isConsult' value='{ $isConsult }'>
122             <input type='submit' value='Go Back' /></a>
123             </form>");
124     }
125     else
126     {
127         ##### SET UP STRINGS TO USE IN INSERTION
128         #####
129         $teenager = "teenager dental charting";
130         $adult = "adult dental charting";
131         $TopR = "top right";
132         $TopL = "top left";
133         $BottomL = "bottom left";
134         $BottomR = "bottom right";

```



```

134
135          #####          FOR ADULT CLIENT
136          #####
137
138      if ($client_age>19)
139      {
140          echo("<p>Searching for procedure.</p>");
141          ## CHECK IF THERE IS A PROCEDURE ALREADY
142          $find_procedure = $connection->prepare("SELECT *
143
144          procedure_in_consultation
145
146
147          WHERE VAT_doctor
148
149          =:VAT_doctor
150
151          AND
152
153          date_timestamp=:date_timestamp
154
155          AND name =:proc")
156
157      ;
158
159      $result3=$find_procedure ->execute([
160      'VAT_doctor' => $VAT_doctor,
161      'date_timestamp'=> $date_timestamp,
162      'proc' => $adult,
163      ]);
164      $n_proc = $find_procedure->rowCount();
165      if ($result3== FALSE)
166      {
167          echo("Couldn't find procedure.");
168          $info = $connection->errorInfo();
169          echo("<p>Error: {$info[1]}</p>");
170          exit();
171      }
172      elseif($n_proc == 0)
173      {
174          echo("<p>Didnt find procedure, so adding it.</p>");
175          ## IF THERE ISNT PROCEDURE, ADD IT
176          $sql = "INSERT INTO procedure_in_consultation VALUES
177          ('$adult', $VAT_doctor, '$date_timestamp', '$description')";
178          $nrows = $connection->exec($sql);
179          if($nrows== FALSE)
180          {
181              echo("<p>Couldn't add procedure_in_consultation
182
183              .</p>");
184
185              $info = $connection->errorInfo();
186              echo("<p>Error: {$info[1]}</p>");
187              exit();
188          }
189          else
190          {
191              echo("<p> procedure added!!</p>");
192              #### ADD INFO FOR EACH TEETH ####
193              $i = 1;
194              foreach ($TR as $key => $value)
195              {
196                  if (!empty($value) && !empty($TRD[$key]))
197                  {
198                      $sql1 = "INSERT INTO procedure_charting
199          VALUES ('$adult', $VAT_doctor, '$date_timestamp', '$TopR', $i, '$TRD
200          [$key]', $value)";

```

```

183         $nrows = $connection->exec($sql1);
184         if($nrows== FALSE)
185         {
186             echo("<p>Couldn't add $TopR $key
teeth information.</p>");
187             $info = $connection->errorInfo();
188             echo("<p>Error: {$info[1]}</p>");
189             exit();
190         }
191     }
192     $i++;
193 }
194 $i = 1;
195 foreach ($TL as $key => $value)
196 {
197     if (!empty($value) && !empty($TLD[$key]))
198     {
199         $sql2 = "INSERT INTO procedure_charting
VALUES ('$adult', $VAT_doctor, '$date_timestamp', '$TopL', $i, '$TLD
[$key]', $value)";
200
201         $nrows = $connection->exec($sql2);
202         if($nrows== FALSE)
203         {
204             echo("<p>Couldn't add $TopL $key
teeth information.</p>");
205             $info = $connection->errorInfo();
206             echo("<p>Error: {$info[1]}</p>");
207             exit();
208         }
209     }
210     $i++;
211 }
212 $i = 1;
213 foreach ($BR as $key => $value)
214 {
215     if (!empty($value) && !empty($BRD[$key]))
216     {
217         $sql3 = "INSERT INTO procedure_charting
VALUES ('$adult', $VAT_doctor, '$date_timestamp', '$BottomR', $i, '
$BRD[$key]', $value)";
218
219         $nrows = $connection->exec($sql3);
220         if($nrows== FALSE)
221         {
222             echo("<p>Couldn't add $BottomR $key
teeth information.</p>");
223             $info = $connection->errorInfo();
224             echo("<p>Error: {$info[1]}</p>");
225             exit();
226         }
227     }
228     $i++;
229 }
230 $i = 1;
231 foreach ($BL as $key => $value)
232 {
233     if (!empty($value) && !empty($BLD[$key]))

```

```

234         {
235             $sql4 = "INSERT INTO procedure_charting
VALUES ('$adult', $VAT_doctor, '$date_timestamp', '$BottomL', $i, '
$BLD[$key]', $value)";
236
237             $nrows = $connection->exec($sql4);
238             if($nrows== FALSE)
239             {
240                 echo("<p>Couldn't add $BottomL $key
teeth information.</p>");
241                 $info = $connection->errorInfo();
242                 echo("<p>Error: {$info[1]}</p>");
243                 exit();
244             }
245         }
246         $i++;
247     }
248 }
249 }
250 }
251 else
252 {
253     echo("<p>Searching for procedure.</p>");
254     ## CHECK IF THERE IS A PROCEDURE ALREADY
255     $find_procedure = $connection->prepare("SELECT *
256                                         FROM
procedure_in_consultation
257                                         WHERE VAT_doctor
=:VAT_doctor
258                                         AND
date_timestamp=:date_timestamp
259                                         AND name =:proc")
;
260     $result3=$find_procedure ->execute([
261     'VAT_doctor' => $VAT_doctor,
262     'date_timestamp'=> $date_timestamp,
263     'proc' => $teenager,
264     ]);
265     $n_proc = $find_procedure->rowCount();
266     if ($result3== FALSE)
267     {
268         echo("Couldn't find procedure.");
269         $info = $connection->errorInfo();
270         echo("<p>Error: {$info[1]}</p>");
271         exit();
272     }
273     elseif($n_proc == 0)
274     {
275         echo("<p>Didnt find procedure, so adding it.</p>");
276         ## IF THERE ISNT PROCEDURE, ADD IT
277         $sql = "INSERT INTO procedure_in_consultation VALUES
('$teenager', $VAT_doctor, '$date_timestamp', '$description')";
278         $nrows = $connection->exec($sql);
279         if($nrows== FALSE)
280         {
281             echo("<p>Couldn't add procedure_in_consultation
.</p>");
282             $info = $connection->errorInfo();

```

```

283         echo("<p>Error: {$info[1]}</p>");
284         exit();
285     }
286     else
287     {
288         echo("<p> procedure added!!</p>");
289         ##### ADD INFO FOR EACH TEETH #####
290         $i = 1;
291         foreach ($TR as $key => $value)
292         {
293             if (!empty($value) && !empty($TRD[$key]))
294             {
295                 $sql1 = "INSERT INTO procedure_charting
VALUES ('$teenager', $VAT_doctor, '$date_timestamp', '$TopR', $i, '
$TRD[$key]', $value)";
296
297                 $nrows = $connection->exec($sql1);
298                 if($nrows== FALSE)
299                 {
300                     echo("<p>Couldn't add $TopR $key
teeth information.</p>");
301
302                     $info = $connection->errorInfo();
303                     echo("<p>Error: {$info[1]}</p>");
304                     exit();
305                 }
306                 $i++;
307             }
308             $i = 1;
309             foreach ($TL as $key => $value)
310             {
311                 if (!empty($value) && !empty($TLD[$key]))
312                 {
313                     $sql2 = "INSERT INTO procedure_charting
VALUES ('$teenager', $VAT_doctor, '$date_timestamp', '$TopL', $i, '
$TLD[$key]', $value)";
314
315                     $nrows = $connection->exec($sql2);
316                     if($nrows== FALSE)
317                     {
318                         echo("<p>Couldn't add $TopL $key
teeth information.</p>");
319
320                         $info = $connection->errorInfo();
321                         echo("<p>Error: {$info[1]}</p>");
322                         exit();
323                     }
324                     $i++;
325                 }
326                 $i = 1;
327                 foreach ($BR as $key => $value)
328                 {
329                     if (!empty($value) && !empty($BRD[$key]))
330                     {
331                         $sql3 = "INSERT INTO procedure_charting
VALUES ('$teenager', $VAT_doctor, '$date_timestamp', '$BottomR', $i,
'$BRD[$key]', $value)";
332

```

```

333         $nrows = $connection->exec($sql3);
334         if($nrows== FALSE)
335         {
336             echo("<p>Couldn't add $BottomR $key
teeth information.</p>");
337             $info = $connection->errorInfo();
338             echo("<p>Error: {$info[1]}</p>");
339             exit();
340         }
341     }
342     $i++;
343 }
344 $i = 1;
345 foreach ($BL as $key => $value)
346 {
347     if (!empty($value) && !empty($BLD[$key]))
348     {
349         $sql4 = "INSERT INTO procedure_charting
VALUES ('$teenager', $VAT_doctor, '$date_timestamp', '$BottomL', $i,
'$BLD[$key]', $value)";
350
351         $nrows = $connection->exec($sql4);
352         if($nrows== FALSE)
353         {
354             echo("<p>Couldn't add $BottomL $key
teeth information.</p>");
355             $info = $connection->errorInfo();
356             echo("<p>Error: {$info[1]}</p>");
357             exit();
358         }
359     }
360     $i++;
361 }
362 }
363 }
364 }
365 header('Location: http://web.tecnico.ulisboa.pt/ist425487/
list_cons_info.php?VAT_doctor='.$VAT_doctor.'&date_timestamp='.
$date_timestamp.'&isConsult='.$isConsult.'&VAT_client='.$VAT_client)
;
366     die();
367 }
368 $connection = null;
369 ?>
370
371 </body>
372 </html>

```