SIBD PROJECT

ASSIGNMENT 3: USING THE DATABASE

COURSE: INFORMATION SYSTEMS AND DATABASES
FACULTY: BRUNO MARTINS

MIGUEL RODRIGUES Nº76176
HENRIQUE PIEDADE Nº75546
RAFAEL VILLAREJO Nº91712
GROUP 63

7/12/2018

1 A Web Application Using the Database

For the web applications, a PHP document was created with all the necessary info to connect to the database. It has to be required in every webpage that wants to use the database:

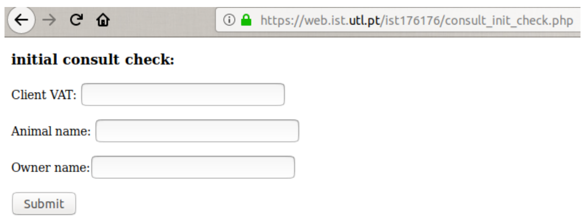
Listing 1: pdo_connection.php

```php
<?php
  $host = "db.tecnico.ulisboa.pt";
  $user = "ist176176";
  $pass = "oalh0726";
  $dsn = "mysql:host=$host;dbname=$user";
  try
  {
    $conn = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception)
  {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }
?>
```

All of the remaining code is present in the annexed files. Throughout the code, PDO prepared statements were used and certain limitations to user inputs where made, e.g.: not having numbers below 0 in forms requiring numbers. Also, there are buttons to return to the previous menu or initial one, so some values are passed in order to ensure the return is made to the correct page (e.g.: passing a client's VAT so that one can return to a consult list).

1.1 Task 1

For the first task, there is a web page (https://web.ist.utl.pt/ist176176/consult_init_check.php) created so that a user can input a client VAT, an animal name, and an owner name. There are no restrictions as to what may be inserted in this page. The inputs are then directed to a page given by the code in *get_consult.php* . Here, in the first place, a check is made to ensure the given VAT corresponds to a client in the system database. If not, it won't allow for nothing else (figure 2). Secondly, checks if there's an animal with the given name in the database and if there's not, the chance to insert a new one assuming that that client is its owner is given by filling a form and calling the code in *insert_new_animal.php* (figure 3).


https://web.ist.utl.pt/ist176176/consult_init_check.php

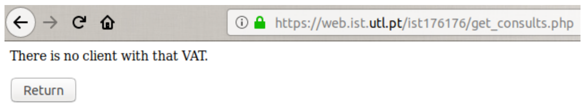
initial consult check:

Client VAT:

Animal name:

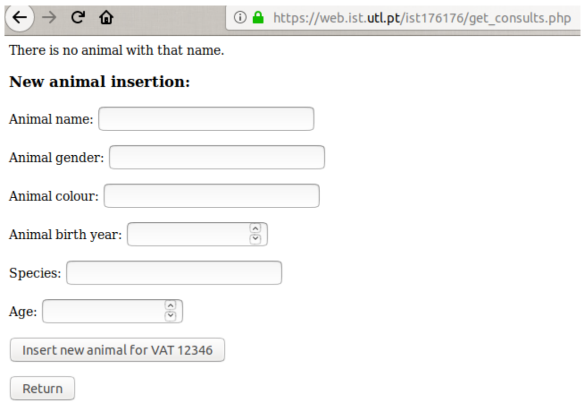
Owner name:

Figure 1: consult_init_check.php



There is no client with that VAT.

Figure 2: Page displayed when the VAT isn't in the Client table.


https://web.ist.utl.pt/ist176176/get_consults.php

There is no animal with that name.

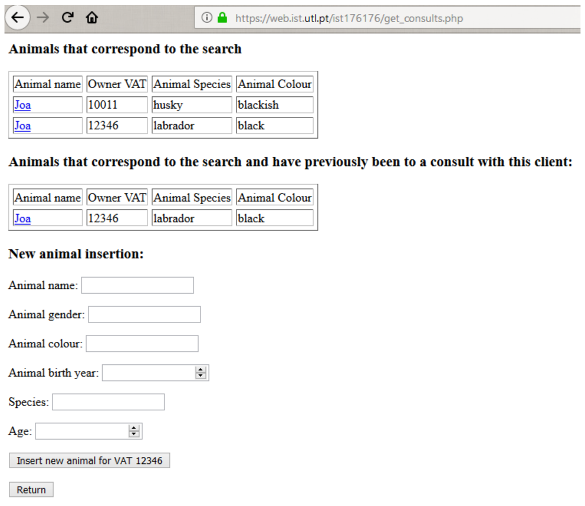
New animal insertion:

Animal name:	
Animal gender:	
Animal colour:	
Animal birth year:	
Species:	
Age:	

Insert new animal for VAT 12346
Return

Figure 3: Page displayed when the animal's name doesn't exist in the database.

If both checks are passed, then all the known animals that correspond to that animal name with its owner name being the one given (or a part of) are shown. In a table below, the animals who correspond to the previous search and have been to a consult with the client that gave his VAT are shown. This is done separately so that there's no omission of the correct animal if it hasn't been to a consult with this client yet and in the case where a lot of animals correspond to the first search, the second table narrows down the hypothesis if the client has already been to a consult with the animal (figure 4. There is no problem in inserting nothing in the owner's name, since sometimes it may not be needed (e.g.: if an animal gets lost but has its name in the collar but the owner is unknown). If the client's VAT exists and the animal's name also exists, but there's no correspondence when the owner's name (or part of) is used, the page displayed is the one if figure 5 where there is a chance to insert an animal, but doesn't show results (since there aren't any).



Animals that correspond to the search

Animal name	Owner VAT	Animal Species	Animal Colour
Joa	10011	husky	blackish
Joa	12346	labrador	black

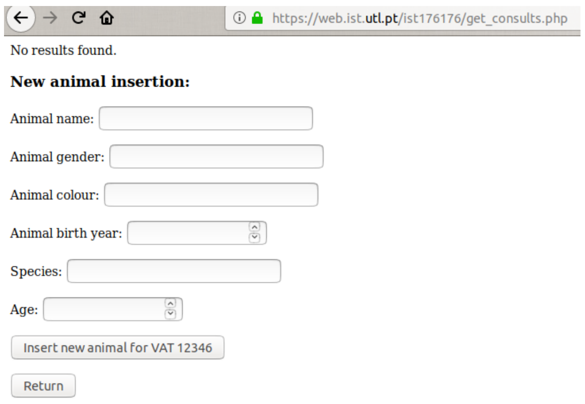
Animals that correspond to the search and have previously been to a consult with this client:

Animal name	Owner VAT	Animal Species	Animal Colour
Joa	12346	labrador	black

New animal insertion:

Animal name:	
Animal gender:	
Animal colour:	
Animal birth year:	
Species:	
Age:	

Figure 4: Page displayed when the search yields results.


https://web.ist.utl.pt/ist176176/get_consults.php

No results found.

New animal insertion:

Animal name:

Animal gender:

Animal colour:

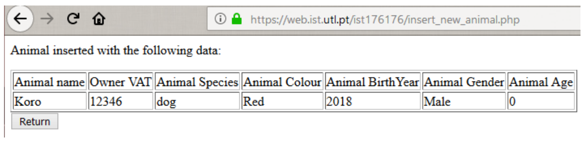
Animal birth year:

Species:

Age:

Figure 5: Page displayed when the search yields no results.

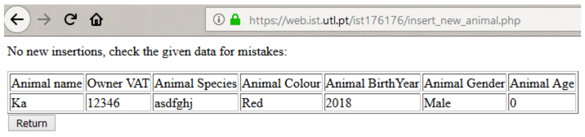
The insertion of a new animal is done through the code in the file *insert_new_animal.php* (to which the second to last button in the previous page points to, passing as arguments the fields in the form and the client's VAT) When inserting a new animal, a result for success or failure is given, along with the given information about the new animal, as described in figures 6 and 7.


https://web.ist.utl.pt/ist176176/insert_new_animal.php

Animal inserted with the following data:

Animal name	Owner VAT	Animal Species	Animal Colour	Animal BirthYear	Animal Gender	Animal Age
Koro	12346	dog	Red	2018	Male	0

Figure 6: Page displayed when the animal insertion succeeds.


https://web.ist.utl.pt/ist176176/insert_new_animal.php

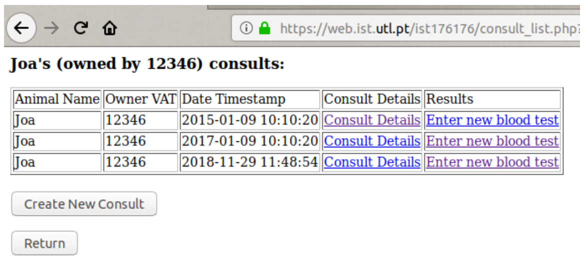
No new insertions, check the given data for mistakes:

Animal name	Owner VAT	Animal Species	Animal Colour	Animal BirthYear	Animal Gender	Animal Age
Ka	12346	asdfghj	Red	2018	Male	0

Figure 7: Page displayed when the animal insertion fails.

1.2 Task 2

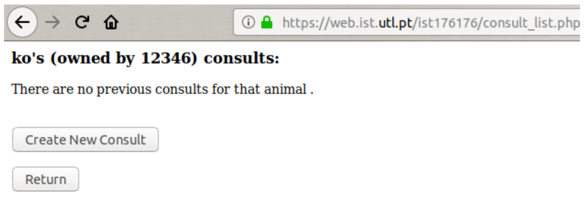
In figure 4, the names of the shown animals can be clicked to go through to a page where a history of consults is shown, as per the code in the file *consult_list.php* which will show figure 8 or 9, whether that animal has previous consults or not, respectively. In both pages there is an option to create a new consult that will call *new_consult.php*, along with animal's name, the client's VAT and the owner's VAT, passed from the previous page.



Joa's (owned by 12346) consults:

Animal Name	Owner VAT	Date Timestamp	Consult Details	Results
Joa	12346	2015-01-09 10:10:20	Consult Details	Enter new blood test
Joa	12346	2017-01-09 10:10:20	Consult Details	Enter new blood test
Joa	12346	2018-11-29 11:48:54	Consult Details	Enter new blood test

Figure 8: Page displayed when the animal has previous consults.

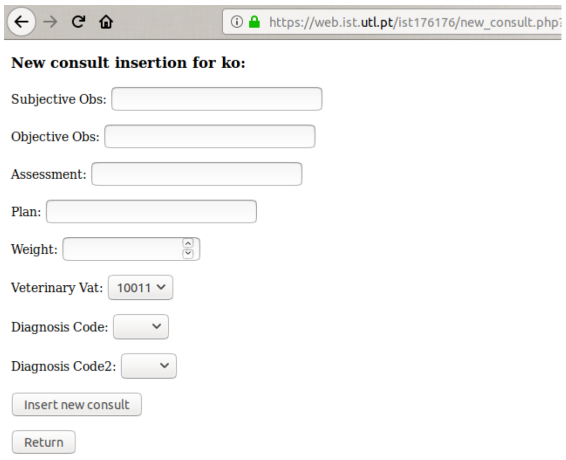


ko's (owned by 12346) consults:

There are no previous consults for that animal .

Figure 9: Page displayed when the animal doesn't have previous consults.

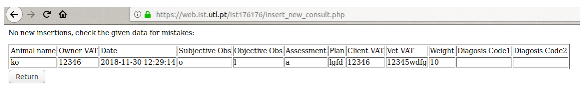
When creating a new consult, one is presented with the page in figure 10 where all the information is filled, including diagnosis codes which can be left blank if there's no need for a diagnosis code. When submitting the information, the user is directed to the page *insert_new_consult.php* where an insertion of the given information is made, unless an error is found and in that case the user is informed of an error - to avoid some, the weight has to be greater than 0.01 and there veterinary VAT field must be filled, hence the dropdown without an empty option. In either cases, error or success, the user is presented with the inserted data in order to check where the mistake is (figure 11) or to check what has been inserted (figure 12). If the consult insertion fails, there's no need to see and use the consult diagnosis inserted.



New consult insertion for ko:

Subjective Obs:	
Objective Obs:	
Assessment:	
Plan:	
Weight:	
Veterinary Vat:	10011
Diagnosis Code:	
Diagnosis Code2:	

Figure 10: Form for a new consult insertion.



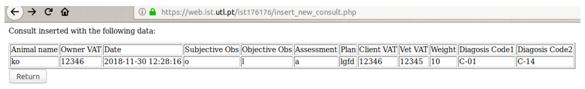
https://web.ist.utl.pt/ist176176/insert_new_consult.php

No new insertions, check the given data for mistakes:

Animal name	Owner VAT	Date	Subjective Obs	Objective Obs	Assessment	Plan	Client VAT	Vet VAT	Weight	Diagosis Code1	Diagosis Code2
ko	12346	2018-11-30 12:29:14	o	l	a	lgfd	12346	12345wdfg	10		

Return

Figure 11: Page displayed when the consult information has a mistake and is not inserted.



https://web.ist.utl.pt/ist176176/insert_new_consult.php

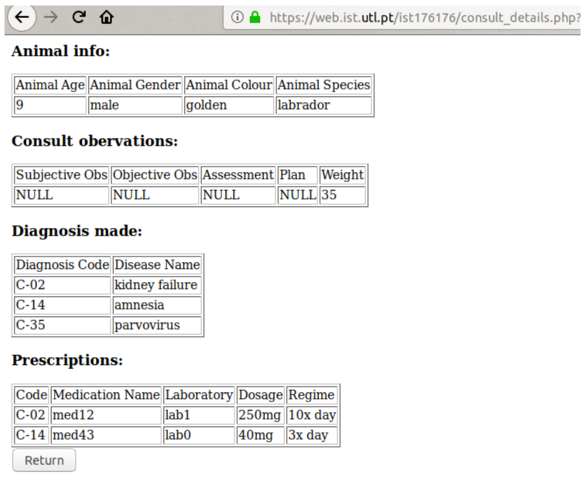
Consult inserted with the following data:

Animal name	Owner VAT	Date	Subjective Obs	Objective Obs	Assessment	Plan	Client VAT	Vet VAT	Weight	Diagosis Code1	Diagosis Code2
ko	12346	2018-11-30 12:28:16	o	l	a	lgfd	12346	12345	10	C-01	C-14

Return

Figure 12: Page displayed when the consult is correctly inserted.

Clicking the "Consult Details" link in figure 8 will lead to a page given by the code in *consult_details.php* where all the specifics about that consult are shown. The animal's name, the owner's VAT, the client's VAT and the timestamp of the selected consult are passed to this page. Several queries are made: first to the animal table where the specific animal info is extracted; then to the consult table where the SOAP notes and weight are; thirdly to both the diagnostic codes and consult diagnosis tables in order to get the diagnosis resulting from this consult; lastly the prescription table is queried in order to show any medication prescribed resulting from this consult and diagnosis. If the first query shows an error, then there's no need to do any other. The same can be said for any query to the following ones, as there are no diagnosis with consults and no prescriptions without diagnosis. There might also be the case where a consult leads to no diagnosis or a diagnosis leads to no prescriptions. In figure 13 an example is shown of the data extracted by this page, where the return button returns to the list of consults instead of to the main page as the others do.



https://web.ist.utl.pt/ist176176/consult_details.php?

Animal info:

Animal Age	Animal Gender	Animal Colour	Animal Species
9	male	golden	labrador

Consult obervations:

Subjective Obs	Objective Obs	Assessment	Plan	Weight
NULL	NULL	NULL	NULL	35

Diagnosis made:

Diagnosis Code	Disease Name
C-02	kidney failure
C-14	amnesia
C-35	parvovirus

Prescriptions:

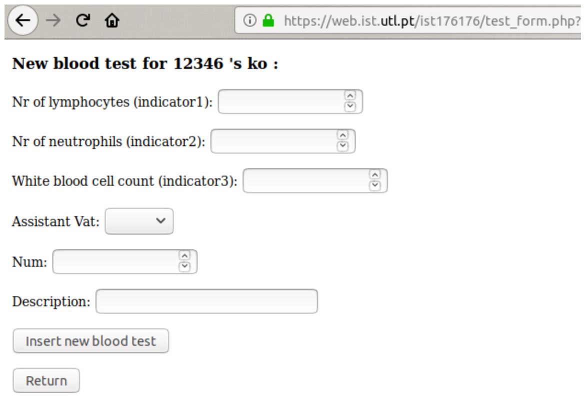
Code	Medication Name	Laboratory	Dosage	Regime
C-02	med12	lab1	250mg	10x day
C-14	med43	lab0	40mg	3x day

Return

Figure 13: Page displaying the details of a given consult.

1.3 Task 3

For the last task, there is another link ("Enter new blood test"), from the same page in figure 8 where a new blood test entry can be made. By clicking it, one is directed to the page given by *test_form.php*, passing the same information as before (animal's name, owner's VAT, client's VAT and timestamp) and is presented with a form for the information to be submitted (figure 14). Here, all the values must be greater or equal to zero (when left blank, the inserted value is zero) and an assistant VAT may be left in blank, since a procedure may not be performed by an assistant. A fixed set of indicators are being used, as described in the handout.



New blood test for 12346 's ko :

Nr of lymphocytes (indicator1):

Nr of neutrophils (indicator2):

White blood cell count (indicator3):

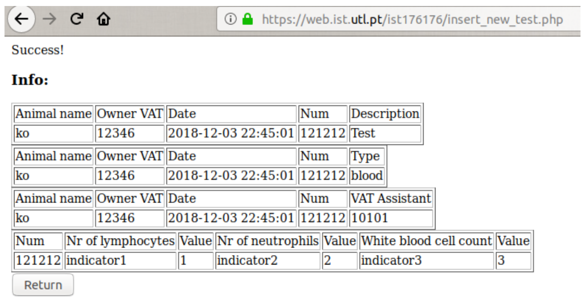
Assistant Vat:

Num:

Description:

Figure 14: Page displaying the form to insert a new blood test.

After filling the table, pressing the "Insert new blood test" button will lead to the code in *insert_new_test.php*, passing all the necessary information and leading to the insertion of the required table entries, which may be three or four, since having an assistant is optional. These are all made in a single transaction, so that if one fails, all are aborted, using the rollback function. before committing, all queries have the number of affected rows counted and if either is zero, an exception is thrown that is caught in the PHP code, triggering the rollback. In case of success (figure 15) or in case of failure (figure 16), the inserted data is shown. The return button leads to the previous consult list.



Success!

Info:

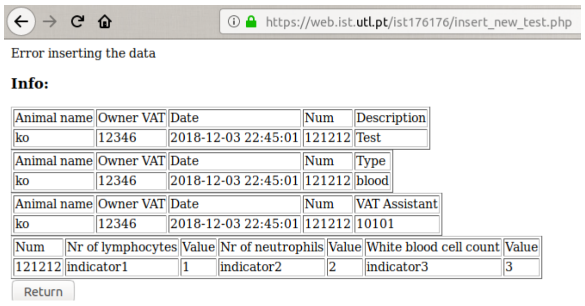
Animal name	Owner VAT	Date	Num	Description
ko	12346	2018-12-03 22:45:01	121212	Test

Animal name	Owner VAT	Date	Num	Type
ko	12346	2018-12-03 22:45:01	121212	blood

Animal name	Owner VAT	Date	Num	VAT Assistant
ko	12346	2018-12-03 22:45:01	121212	10101

Num	Nr of lymphocytes	Value	Nr of neutrophils	Value	White blood cell count	Value
121212	indicator1	1	indicator2	2	indicator3	3

Figure 15: Correct insertion of a new blood test.



https://web.ist.utl.pt/ist176176/insert_new_test.php

Error inserting the data

Info:

Animal name	Owner VAT	Date	Num	Description
ko	12346	2018-12-03 22:45:01	121212	Test

Animal name	Owner VAT	Date	Num	Type
ko	12346	2018-12-03 22:45:01	121212	blood

Animal name	Owner VAT	Date	Num	VAT Assistant
ko	12346	2018-12-03 22:45:01	121212	10101

Num	Nr of lymphocytes	Value	Nr of neutrophils	Value	White blood cell count	Value
121212	indicator1	1	indicator2	2	indicator3	3

Return

Figure 16: Failure inserting a new blood test.

2 Functions, Triggers and Stored Procedures

2.1 Triggers

```
--1

delimiter $$
create trigger act_age after insert on consult
for each row
begin
  update animal
    set age = year(new.date_timestamp) − birth_year
    where animal.VAT = new.VAT_owner
    and animal.name = new.name;
end$$

delimiter ;
```

For the following triggers, as MySQL only allows for triggers to act before or after an action, the solution found was to send a signal that would abort the operation and print a suitable message (instead of doing an illegal action, that would also abort, as trying to insert a NULL VAT, for simplicity). This way, when the condition is found, the insertion is aborted. It is needed to have triggers for both the insertions and the updates. And in the veterinary/assistant case, if one was to change profession, it would have to be deleted from the former profession's table and only then inserted in the new one.

```
--2

delimiter $$

create trigger check_vet before insert on assistant
 for each row
 begin
 if exists (select * from veterinary v where v.VAT =new.VAT) then
   signal sqlstate '45000'
   set message_text = "There's already a veterinary with that VAT.";
 end if;
end $$

delimiter ;

delimiter $$

create trigger check_vet2 before update on assistant
 for each row
 begin
 if exists (select * from veterinary v where v.VAT =new.VAT) then
   signal sqlstate '45000'
   set message_text = "There's already a veterinary with that VAT.";
 end if;
end $$

delimiter ;
delimiter $$

create trigger  check_assist before insert on veterinary
 for  each row
 begin
 if  exists  (select  * from assistant a where a.VAT =new.VAT) then
   signal sqlstate  '45000'
   set  message_text = "There's already an assistant with that VAT.";
  end if;
end $$

delimiter ;

delimiter $$

create trigger  check_assist2 before update on veterinary
 for  each row
 begin
 if  exists  (select  * from assistant a where a.VAT =new.VAT) then
   signal sqlstate  '45000'
   set  message_text = "There's already an assistant with that VAT.";
  end if;
end $$

delimiter ;

--3

delimiter $$

create trigger  check_indiv before insert on phone_number
 for  each row
 begin
 if  exists(select  * from phone_number pn where new.phone = pn.phone) then
   signal sqlstate  '45000'
   set  message_text = "That phone number is already associated with another individual.";
 end if;
end$$

delimiter ;

delimiter $$

create trigger  check_indiv2 before update on phone_number
 for  each row
 begin
 if  exists(select  * from phone_number pn where new.phone = pn.phone) then
   signal sqlstate  '45000'
   set  message_text = "That phone number is already associated with another individual.";
 end if;
end$$

delimiter ;
```

2.2 Function

In order to get the number of consults a specific animal had in a specific year, one also needs to provide the owner's VAT, as different animals can have the same name.

```
delimiter $$

create function count_consults(a_name varchar(255), a_VAT varchar(25), y integer)
 returns integer
 begin
 declare total  integer;
 select  count(*) into  total
 from consult c
 where c.name = a_name
 and c.VAT_owner = a_VAT
 and year(c.date_timestamp) = y;
 return total;
end$$

delimiter ;
```

2.3 Stored Procedure

```
delimiter $$

create procedure change_val()
  begin
  update produced_indicator p
    set  p.value = 0.1*p.value,
    p.name = p.name,
    p.VAT_owner = p.VAT_owner,
    p.date_timestamp = p.date_timestamp,
    p.num = p.num,
    p.indicator_name = p.indicator_name
    where p.indicator_name in(
      select  indicator.name
      from indicator
      where indicator.units = 'milligrams');

  update indicator i
    set  i.units = 'centigrams',
    i.reference_value = 0.1*i.reference_value
    where i.units= 'milligrams';
  end $$

delimiter ;
```