Instituto Superior Técnico

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Maria Margarida Dias dos Reis	n.º 73099
Tiago José Ribeiro Ricardo	n.º 73649
Sofia Fidalgo da Silva	n.º 73483

Grupo n.º 16

Lisboa, 25 de Novembro de 2015

1 DataBase Creation

The SQL instructions written to create the database described in the relational model of the project are in the following script:

```sql
drop table if exists Wears;
drop table if exists Lives;
drop table if exists Connects;
drop table if exists Reading;
drop table if exists Setting;
drop table if exists Actuator;
drop table if exists Sensor;
drop table if exists Patient;
drop table if exists PAN;
drop table if exists Device;
drop table if exists Municipality;
drop table if exists Period;

create table Patient
   (number  integer(9), -- health no. in portugal has 9 digits
    name    varchar(255) NOT NULL,
    address varchar(255),
    primary key(number));

create table PAN
   (domain  varchar(255),
    phone   integer(9) NOT NULL,  -- phone numbers in portugal have 9 digits
    primary key(domain));

create table Device
   (serialnum     numeric(8,0), -- how many digits has the serial number of a device?
    manufacturer  varchar(255),
    description   varchar(255) NOT NULL , -- it is stated that the Devices must have
    a Description
    primary key(serialnum, manufacturer));

create table Sensor
   (snum  numeric(8,0),
    manuf varchar(255),
    units varchar(255) NOT NULL, -- number of units cannot be NULL, if none it should
     be 0
    primary key(snum, manuf),
    foreign key(snum, manuf) references Device(serialnum, manufacturer));

create table Actuator
   (snum  numeric(8,0),
    manuf varchar(255),
    units varchar(255) NOT NULL, -- number of units cannot be NULL
```

```
42      primary key(snum, manuf),
43      foreign key(snum, manuf) references Device(serialnum, manufacturer));
44
45  create table Municipality
46     (nut4code  integer(5), -- 5 digit code assigned by the National Bureau of
       Statistics
47      name       varchar(255) NOT NULL, -- municipality must have a name
48      primary key(nut4code));
49
50  create table Period
51     (start datetime,
52      end   datetime,
53      primary key(start, end));
54
55  create table Reading
56     (snum       numeric(8,0),
57      manuf      varchar(255),
58      datetime   datetime,
59      value      numeric(5,2) NOT NULL,  -- values read from the sensors have 5 total
       digits and 2 fractional digits
60      primary key(snum, manuf, datetime),
61      foreign key(snum, manuf) references Sensor(snum, manuf));
62
63  create table Setting
64     (snum       numeric(8,0),
65      manuf      varchar(255),
66      datetime   datetime,
67      value      numeric(5,2), -- settings sent to the actuators have 5 total digits and
        2 fractional digits
68      primary key(snum, manuf, datetime),
69      foreign key(snum, manuf) references Actuator(snum, manuf));
70
71  create table Wears
72     (start    datetime,
73      end      datetime,
74      patient  integer(9),
75      pan      varchar(255),
76      primary key(start, end, patient),
77      foreign key(start, end) references Period(start, end),
78      foreign key(patient) references Patient(number),
79      foreign key(pan) references PAN(domain));
80
81  create table Lives
82     (start    datetime,
83      end      datetime,
84      patient  integer(9),
85      muni     integer(5),
```

```
    primary key(start, end, patient),
    foreign key(start, end) references Period(start, end),
    foreign key(patient) references Patient(number),
    foreign key(muni) references Municipality(nut4code));

create table Connects
   (start datetime,
    end   datetime,
    snum  numeric(8,0),
    manuf varchar(255),
    pan   varchar(255),
    primary key(start, end, snum, manuf),
    foreign key(start, end) references Period(start, end),
    foreign key(snum, manuf) references Device(serialnum, manufacturer),
    foreign key(pan) references PAN(domain));
```

1.a Additional Notes

1.a.1 Patient

The patient should have a name, so the field name is NOT NULL and the identification number has 9 digits only (a trigger was developed to guarantee that the user can't add numbers in the field with less than 9 digits).

1.a.2 PAN

The domain of the PAN is a variable of type varchar(255) and the phone number associated to it is an integer of 9 digits (a trigger was also developed to guarantee that the user can't add phone numbers in the field with less than 9 digits).

1.a.3 Device, Sensor and Actuator

The serial number of the device is a number with 8 digits and besides having a manufacturer and it should have a description, considered in the Queries section (a trigger was also developed to guarantee that the user can't add serial numbers in the field with less than 8 digits).

1.a.4 Municipality

The nut4code is an integer with 5 digits (a trigger was also developed to guarantee that the user can't add nut4codes in the field with more or less than 5 digits).

1.a.5 Period

Fields start and end of table Period are of type datetime instead of timestamp because in the second case the range only goes from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC. Devices and

PANs that are presently connected have as end date 2999-12-31 00:00:00, which is only possible with the range of datetime.

1.a.6 Reading and Setting

The datetime of Setting and Reading is also of type datetime.

2 Triggers

While associating devices/patients to a PAN, one may consider overlapping periods that aren't possible due to the project specifications. In order to prevent these events, several triggers were written with respect to the tables Period, Wears and Connects.

2.a Table Period

The periods associated to the connection of devices and patients to PANs should be consistent: the start date needs to be smaller than the end date. The first trigger (`check_valid_period_i`), associated with the table Period, was written to prevent the user from inserting periods into the table in which the start date occurs after than the end date. If this happens, an error (`period_not_valid()`) is thrown.

```
delimiter $$
create trigger check_valid_period_i before insert on Period
  for each row
  begin
  #end tem de ser maior que start
    if new.start>new.end then
    call period_not_valid();
  end if;
end$$
delimiter ;
```

2.b Table Wears

In the table Wears two issues need to be addressed: a Patient can only be connected to one PAN during a specific time period and *vice-versa*. If patient '123' is wearing PAN1 during the time interval $[t_i, t_f]$, one must prevent the connection of another PAN to that patient during an intersecting time interval and also to prevent the association of PAN1 to another patient during that period. In order to do so, two triggers were wrote for both the INSERT and the UPDATE statements in MySQL.

2.b.1 INSERT

For the INSERT statement it is required to prevent the user from inserting a new period that intersects the existing associated periods in two different aspects:

- The patient cannot be connected to 2 PANs at the same time interval;
- The same PAN cannot be connected at 2 patients at the same time interval. The trigger for the INSERT statement consists therefore in six counters that address the following period intersections:
- The new start time instant is between a period $[t_i, t_f]$ in which the patient or the PAN is already associated;
- The new end time instant is between a period $[t_i, t_f]$ in which the patient or the PAN are already associated;
- The new time period is totally contained inside a period $[t_i, t_f]$ in which the patient or the PAN are already associated (new $t_i < t_i$ AND new $t_f > t_f$).

If a PAN is associated in any of these situations, the error `another_patient_has_that_PAN_in_that_period()` is thrown. If a patient is associated in any of the situations the error `that_patient_has_a_PAN_in_that_period()` is thrown.

```
delimiter $$
create trigger check_valid_wears_i before insert on Wears
  for each row
  begin
    declare count_patient_1 integer;
    declare count_patient_2 integer;
    declare count_patient_3 integer;
    declare count_pan_1 integer;
    declare count_pan_2 integer;
    declare count_pan_3 integer;

    select count(*) into count_patient_1 from Wears where new.patient = patient and
    new.start between start and end;
    select count(*) into count_patient_2 from Wears where new.patient = patient and
    new.end between start and end;
    select count(*) into count_patient_3 from Wears where new.patient = patient and
    new.start < start and new.end > end;

    select count(*) into count_pan_1 from Wears where new.pan = pan and new.start
    between start and end;
  select count(*) into count_pan_2 from Wears where new.pan = pan and new.end between
     start and end;
    select count(*) into count_pan_3 from Wears where new.pan = pan and new.start <
    start and new.end > end;

    if (count_patient_1 or count_patient_2 or count_patient_3 >= 1) then
      call that_patient_has_a_PAN_in_that_period();
    end if;
```

```
    if (count_pan_1 or count_pan_2 or count_pan_3 >= 1) then
      call another_patient_has_that_PAN_in_that_period();
    end if;
end$$
delimiter ;
```

2.b.2 UPDATE

For the UPDATE statement the following situations need to be taken into account:

- If the new time period is contained inside the old time period, a new PAN can only be associated if it is not associated to other patient in that time interval;
- If the new start time instant is previous to the old start time instant and the new end instant is between the old period, it is just needed to take into account intersections with the new start period;
- If the new time period contains the old time period, it is needed to take into account intersections between the new time instants (start and end) and other periods;
- If the new time period does not intersect the old period, it is required to analyze all intersections.
- If the new end time instant is bigger than the old end time instant and the new start instant is between the old period, it is just needed to take into account intersections with the new old period.

If a PAN is associated in any of these situations, the error `another_patient_has_that_PAN_in_that_period()` is thrown. If a patient is associated in any of the situations the error `that_patient_has_a_PAN_in_that_period()` is thrown.

```
delimiter $$
create trigger check_valid_wears_u before update on Wears
  for each row
  begin
    declare count_patient_1 integer;
    declare count_patient_2 integer;
    declare count_patient_3 integer;
    declare count_pan_1 integer;
    declare count_pan_2 integer;
    declare count_pan_3 integer;


        if(new.start<old.start and new.end<=old.end and new.end>=old.start) then
      select count(*) into count_patient_1 from Wears where new.patient = patient and
     new.start between start and end;
            select count(*) into count_pan_1 from Wears where new.pan = pan and new.
   start between start and end;
```

```
16            if(count_patient_1<>0) then
17        call that_patient_is_connected_to_a_PAN_in_that_period();
18      end if;
19            if (count_pan_1 <> 0) then
20        call another_patient_has_that_PAN_in_that_period();
21      end if;
22    end if;
23
24        if(new.start<old.start and new.end>old.end) then
25      select count(*) into count_patient_1 from Wears where new.patient = patient and
     new.start between start and end;
26      select count(*) into count_patient_2 from Wears where new.patient = patient and
     new.end between start and end;
27      select count(*) into count_pan_1 from Wears where new.pan = pan and new.start
    between start and end;
28      select count(*) into count_pan_2 from Wears where new.pan = pan and new.end
    between start and end;
29
30            if (count_patient_1 or count_patient_2 >= 1) then
31        call that_patient_is_connected_to_a_PAN_in_that_period();
32      end if;
33            if (count_pan_1 or count_pan_2 >= 1) then
34        call another_patient_has_that_PAN_in_that_period();
35      end if;
36    end if;
37
38        if(new.start<old.start and new.end<old.start) then
39      select count(*) into count_patient_1 from Wears where new.patient = patient and
     new.start between start and end;
40      select count(*) into count_patient_2 from Wears where new.patient = patient and
     new.end between start and end;
41      select count(*) into count_patient_3 from Wears where new.patient = patient and
     new.start < start and new.end > end;
42      select count(*) into count_pan_1 from Wears where new.pan = pan and new.start
    between start and end;
43      select count(*) into count_pan_2 from Wears where new.pan = pan and new.end
    between start and end;
44      select count(*) into count_pan_3 from Wears where new.pan = pan and new.start <
     start and new.end > end;
45
46      if (count_pan_1 or count_pan_2 or count_pan_3 >= 1) then
47        call another_patient_has_that_PAN_in_that_period();
48      end if;
49      if (count_patient_1 or count_patient_2 or count_patient_3 >= 1) then
50        call that_patient_is_connected_to_a_PAN_in_that_period();
51            end if;
52    end if;
```

```
53
54        if(new.start>=old.start and new.start>old.end and new.end>old.end) then
55      select count(*) into count_patient_1 from Wears where new.patient = patient and
     new.start between start and end;
56      select count(*) into count_patient_2 from Wears where new.patient = patient and
     new.end between start and end;
57      select count(*) into count_patient_3 from Wears where new.patient = patient and
     new.start < start and new.end > end;
58      select count(*) into count_pan_1 from Wears where new.pan = pan and new.start
    between start and end;
59      select count(*) into count_pan_2 from Wears where new.pan = pan and new.end
    between start and end;
60      select count(*) into count_pan_3 from Wears where new.pan = pan and new.start <
     start and new.end > end;
61
62      if (count_pan_1 or count_pan_2 or count_pan_3 >= 1) then
63        call another_patient_has_that_PAN_in_that_period();
64      end if;
65      if (count_patient_1 or count_patient_2 or count_patient_3 >= 1) then
66        call that_patient_is_connected_to_a_PAN_in_that_period();
67            end if;
68    end if;
69
70        if(new.start>=old.start and new.start<=old.end and new.end>old.end) then
71      select count(*) into count_patient_2 from Wears where new.patient = patient and
     new.end between start and end;
72            select count(*) into count_pan_2 from Wears where new.pan = pan and new.
    end between start and end;
73      if (count_pan_2 <> 0) then
74        call another_patient_has_that_PAN_in_that_period();
75      end if;
76      if(count_patient_2<>0) then
77        call that_patient_is_connected_to_a_PAN_in_that_period();
78      end if;
79    end if;
80
81        if (new.start=old.start and new.start=old.start) then
82      select count(*) into count_pan_1 from Wears where new.pan = pan and new.start
    between start and end;
83      select count(*) into count_pan_2 from Wears where new.pan = pan and new.end
    between start and end;
84      select count(*) into count_pan_3 from Wears where new.pan = pan and new.start <
     start and new.end > end;
85
86      if (count_pan_1 or count_pan_2 or count_pan_3 >= 1) then
87        call another_patient_has_that_PAN_in_that_period();
88      end if;
```

```
    end if;
end$$
delimiter ;
```

2.c Table Connects

In the table Connects the same logic used before was taken into account but for only one issue: a Device can only be connected to one PAN during a specific time period. If device '999' is connected to PAN1 during the time interval $[t_i, t_f]$, one must prevent the connection of this device to another PAN during an intersecting time interval. In order to do so, two triggers were wrote for both the INSERT and the UPDATE statements in MySQL. If the Device is associated to another PAN in the new time period inserted/updated, an error `that_device_is_connected_to_a_PAN_in_that_period()` is thrown.

2.c.1 INSERT

```
delimiter $$
create trigger check_valid_connects_i before insert on Connects
  for each row
  begin
    declare count_condition_1 integer;
    declare count_condition_2 integer;
    declare count_condition_3 integer;

    select count(*) into count_condition_1 from Connects where new.snum = snum and
    new.manuf = manuf and new.start between start and end;
    select count(*) into count_condition_2 from Connects where new.snum = snum and
    new.manuf = manuf and new.end between start and end;
    select count(*) into count_condition_3 from Connects where new.snum = snum and
    new.manuf = manuf and new.start < start and new.end > end;

    if (count_condition_1 or count_condition_2 or count_condition_3 >= 1) then
      call that_device_is_connected_to_a_PAN_in_that_period();
    end if;
end$$
delimiter ;
```

2.c.2 UPDATE

```
delimiter $$
create trigger check_valid_connects_u before update on Connects
  for each row
  begin
    declare count_condition_1 integer;
    declare count_condition_2 integer;
    declare count_condition_3 integer;
```

```
8
9
10        if(new.start<old.start and new.end<=old.end and new.end>=old.start) then
11      select count(*) into count_condition_1 from Connects where new.snum = snum and
    new.manuf = manuf and new.start between start and end;
12      if(count_condition_1<>0) then
13        call that_device_is_connected_to_a_PAN_in_that_period();
14      end if;
15    end if;
16
17        if(new.start<old.start and new.end>old.end) then
18      select count(*) into count_condition_1 from Connects where new.snum = snum and
    new.manuf = manuf and new.start between start and end;
19      select count(*) into count_condition_2 from Connects where new.snum = snum and
    new.manuf = manuf and new.end between start and end;
20
21            if (count_condition_1 or count_condition_2 >= 1) then
22        call that_device_is_connected_to_a_PAN_in_that_period();
23      end if;
24
25    end if;
26
27        if(new.start<old.start and new.end<old.start) then
28      select count(*) into count_condition_1 from Connects where new.snum = snum and
    new.manuf = manuf and new.start between start and end;
29      select count(*) into count_condition_2 from Connects where new.snum = snum and
    new.manuf = manuf and new.end between start and end;
30      select count(*) into count_condition_3 from Connects where new.snum = snum and
    new.manuf = manuf and new.start < start and new.end > end;
31
32
33      if (count_condition_1 or count_condition2 or count_condition_3 >= 1) then
34        call that_device_is_connected_to_a_PAN_in_that_period();
35            end if;
36    end if;
37
38        if(new.start>=old.start and new.start>old.end and new.end>old.end) then
39      select count(*) into count_condition_1 from Connects where new.snum = snum and
    new.manuf = manuf and new.start between start and end;
40      select count(*) into count_condition_2 from Connects where new.snum = snum and
    new.manuf = manuf and new.end between start and end;
41      select count(*) into count_condition_3 from Connects where new.snum = snum and
    new.manuf = manuf and new.start < start and new.end > end;
42
43
44      if (count_condition_1 or count_condition_2 or count_condition_3 >= 1) then
45        call that_device_is_connected_to_a_PAN_in_that_period();
```

```
46              end if;
47      end if;
48
49          if(new.start>=old.start and new.start<=old.end and new.end>old.end) then
50        select count(*) into count_condition_2 from Connects where new.snum = snum and
      new.manuf = manuf and new.end between start and end;
51
52        if(count_condition_2<>0) then
53          call that_device_is_connected_to_a_PAN_in_that_period();
54        end if;
55      end if;
56  end$$
57  delimiter ;
```

3 Queries

3.a

```
1  SELECT DISTINCT r.snum, r.manuf, r.datetime, value, units
2  FROM Reading as r, Sensor as s, Wears as w, Connects as c, Device as d
3  WHERE description = 'blood pressure'
4  AND c.pan = w.pan
5  AND c.snum = d.serialnum
6  AND c.manuf = d.manufacturer
7  AND d.serialnum = s.snum
8  AND d.manufacturer = s.manuf
9  AND s.snum = r.snum
10 AND s.manuf = r.manuf
11 AND patient = '123456789'
12 AND r.datetime BETWEEN c.start AND c.end
13 AND r.datetime BETWEEN w.start AND w.end
14 AND (TIMESTAMPDIFF(MONTH,r.datetime, NOW()) < 6);
```

This query is destined to show all the readings of a patient identified by his/her number in the last 6 months from devices with description 'blood pressure'.

Given this the developed query displays the date, value and units of the readings and the serial number and manufacturer of the corresponding sensor to each of the readings. In lines 12 and 13 are conditions so that we only get readings that correspond to that patient, and not readings from a sensor that was once connected to a PAN that was once worn by that patient, so one must check if the reading was made in a date inside the period when the corresponding device was connected to a PAN and in the period when that PAN was connected to the patient. The patient considered in the example had the number 123456789, and the resulting table is presented in following figure.

snum	manuf	datetime	value	units
98765432	Philips	2015-10-01 08:00:15	60.00	mm Hg
98765432	Philips	2015-10-05 08:00:15	60.30	mm Hg
98765432	Philips	2015-10-09 08:00:15	60.10	mm Hg

3.b

```
1  SELECT DISTINCT muni
2  FROM Lives as l, Connects as c, Wears as w
3  WHERE c.manuf = 'Philips'
4  AND l.patient = w.patient
5  AND w.pan = c.pan
6  AND NOW() BETWEEN w.start AND w.end
7  AND NOW() BETWEEN c.start AND c.end
8  AND NOW() BETWEEN l.start AND l.end
9  GROUP BY muni
10 HAVING COUNT(snum)>= all (SELECT COUNT(snum)
11                       FROM Lives as l1, Connects as c1, Wears as w1
12                       WHERE c1.manuf = 'Philips'
13                       AND l1.patient = w1.patient
14                       AND w1.pan = c1.pan
15                       AND NOW() BETWEEN w1.start AND w1.end
16                       AND NOW() BETWEEN c1.start AND c1.end
17                       AND NOW() BETWEEN l1.start AND l1.end
18                       GROUP BY muni);
```

As for the second query, it is destined to show which is the municipality that currently has the most devices installed from the manufacturer 'Philips'.

The above query displays the 5 digit code that identifies the municipality that fits the above mentioned aspects. For this query it also necessary to verify if the device is currently connected to a PAN that is currently worn by a patient that currently lives in that municipality, and that is checked in the lines 15 to 17. The resulting table is shown in the next figure.

muni
12345

3.c

```
1  SELECT DISTINCT manufacturer
```

```
2  FROM Device as d
3  WHERE description = 'scale'
4  AND NOT EXISTS (SELECT nut4code
5                  FROM Municipality
6                  WHERE nut4code NOT IN(SELECT muni
7                                        FROM Lives as l, Wears as w, Connects as c,
       Device as d1
8                                        WHERE l.patient = w.patient
9                                        AND w.pan = c.pan
10                                       AND c.manuf = d1.manufacturer
11                                       AND c.snum = d1.serialnum
12                                       AND TIMESTAMPDIFF(YEAR,l.end, NOW()) < 1
13                                       AND TIMESTAMPDIFF(YEAR,w.end, NOW()) < 1
14                                       AND TIMESTAMPDIFF(YEAR,c.end, NOW()) < 1
15                                       AND TIMESTAMPDIFF(SECOND,c.start, l.end) >= 0
16                                       AND TIMESTAMPDIFF(SECOND,l.start, c.end) >= 0
17                                       AND TIMESTAMPDIFF(SECOND,l.start, w.end) >= 0
18                                       AND TIMESTAMPDIFF(SECOND,w.start, l.end) >= 0
19                                       AND TIMESTAMPDIFF(SECOND,w.start, c.end) >= 0
20                                       AND TIMESTAMPDIFF(SECOND,c.start, w.end) >= 0
21                                       AND d.manufacturer = d1.manufacturer));
```

Regarding the third query, it is destined to show the manufacturers that had devices described as 'scale' that were worn by patients in all municipalities during the previous year.

As it is hard to convert this sentence into a query, one must reformulate it. With the query, one wants to know what are the manufacturers, that had devices described as 'scale', for which there is no municipality which is not in the set of municipalities where they had devices described as 'scale' being worn in the last year, and for that a nested query is needed. The developed query displays such manufacturers. As done in the previous queries, it is necessary to check if the periods in which a device was connected to a PAN and that PAN was won by a patient and that patient lives in a municipality must interconnect and that is achieved with the lines 15 to 20. The following figure presents the result of the query. NOTE: The tables used in this report are in Section Tables

manufacturer
Xiaomi

4 PHP and HTML

The web application was made to perform two tasks - access patient records and transfer devices between PANs. A simple index page - `index.html` - was created to choose either one of those tasks.

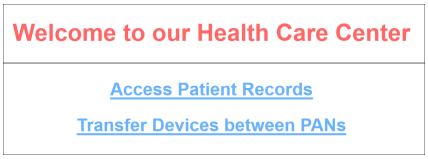


Figure 1: `index.html`.

4.a

Given a patient name, we should display all the readings and settings of a patient (indicating time and date, device, value and units of reading/setting).

To search for a certain patient, one has to input the patient name in a text input field, as can be seen in Figure 2(a) - `patient_records.php`. It is possible to search for characters within a certain name, making it easier for the user that is interacting with the web application, because he does not have to search for the entire patient name. In Figure 2(b) all patients with the letter M are shown, just as the user wanted. Data coming from the text input field is passed onto another web page - `select_patient_records.php` - via a form with `POST` method.

Because the primary key for a patient is the number and not name, it is possible for two or more patients to have the same name. To solve this problem, if two patients have the same name, the option to display the records for all those patients is shown.

If nothing is inserted in the text input field and the submit button is pressed, all patients registered in the health care center are shown, as can be seen in Figure 2(c).

Utilizing the link on the left column of the table it is possible to access the medical records of the desired patient. Data (patient name and number) is passed from this web page to the next one that actually displays the readings and settings - `records.php` - through the URL associated with the link that reads "Show records": `href="records.php?number=[`*insert patient number*`]&name=[`*insert patient name*`]`.

The information is then processed and the readings/settings are shown in separate tables, as can be seen in Figure 3(a). The query that performs this operation is:

```
SELECT DISTINCT Reading.snum, Reading.manuf, Reading.value, Sensor.units, Reading.
    datetime
FROM Patient, Wears, Connects, Reading, Sensor
WHERE Wears.patient = '$number'
  AND Wears.pan = Connects.pan
  AND Connects.snum = Reading.snum
  AND Connects.manuf = Reading.manuf
  AND Sensor.snum = Reading.snum
  AND Sensor.manuf = Reading.manuf
  AND Reading.datetime BETWEEN Connects.start AND Connects.end
  AND Reading.datetime BETWEEN Wears.start AND Wears.end
```

If a certain patient does not have registered readings/settings that information is shown, as seen in Figure 3(b).

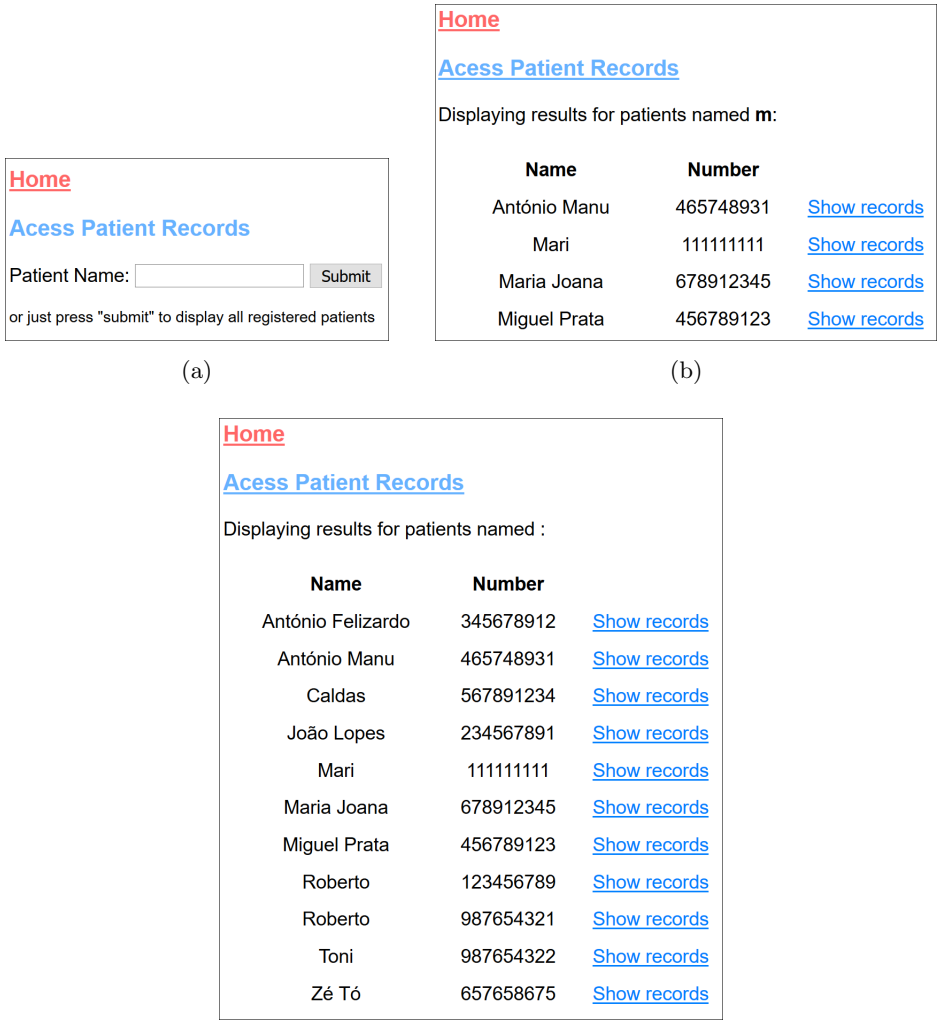


Figure 2: Options to select patients registered in the health care center - `patient_records.php` (a) and `select_patient_records.php` (b), (c).

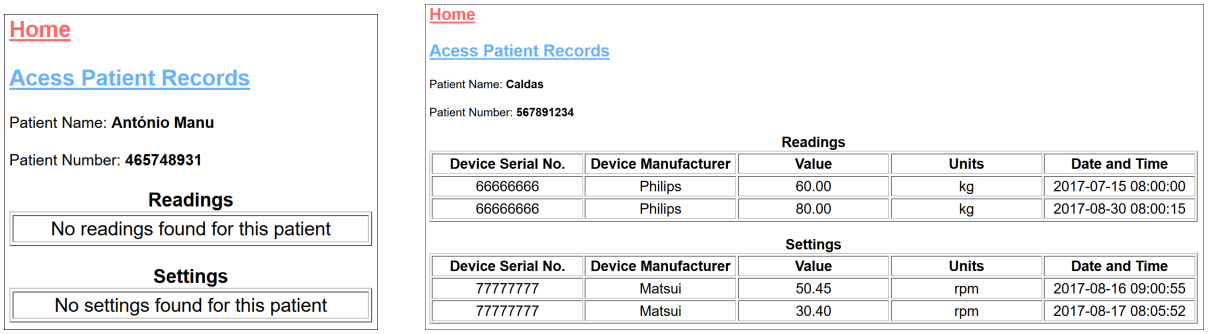
4.b

It is intended to transfer devices from one pan to another, by select a certain patient through his/her name. Let's say we intend to transfer device D previously connected to PAN P1 to PAN P2. Firstly, we have to update the connection period of D to P1 from [start date, 2999-12-31 00:00:00] to [start date, now]. Then we add the connection period of D to P2 as [now, 2999-12-31 00:00:00].

The process to select a patient is the same as described previously, but now for the web pages `transfer_devices.php` and `select_patient_transfer.php`, that also utilize the same methods to get information through.

After selecting the patient, the web page `transfer.php` is presented. Firstly it is shown the patient name, number, his/her current PAN and previous PAN.

To see what is the current PAN from a certain patient it is necessary to find the one registered for that patient in table `Wears` with end date equal to `2999-12-31 00:00:00`, the date that represents



(a) (b)

Figure 3: Accessing patient records - `records.php` (a), (b).

present moments in our database. This query is:

```
SELECT Wears.pan
FROM Wears
WHERE Wears.patient = '$number'
  AND Wears.end = '2999-12-31 00:00:00'
```

If there is no current PAN for the selected patient then the web page seen in Figure 4(a) is presented.

To determine what the previous PAN is, data from table `Wears` is ordered in descending order by the end date of connection to a certain PAN. Then, the second line coming from that query is read, utilizing the `LIMIT` element.

```
SELECT Wears.pan, Wears.end
FROM Wears
WHERE Wears.patient = '$number'
ORDER BY Wears.end DESC
LIMIT 1,1
```

If a certain patient has a current PAN but not a previous PAN, then the web page seen in Figure 4(b) is presented.

After ascertaining what the previous PAN it is necessary to check if such PAN has devices that can be transferred. We have to verify that the devices were connected to the previous PAN during the same time that the patient was connected to it. Also, it is necessary to check that the devices are still connected to the previous PAN.

```
SELECT Connects.snum, Connects.manuf, Connects.start
FROM Connects
WHERE Connects.pan = '$previousPAN'
  AND Connects.start < '$previousPAN_end'
  AND Connects.end = '2999-12-31 00:00:00'
```

If the devices connected to the previous PAN do not meet previous constraints then they are no transferable devices and the web page seen in Figure 4(c) is presented.

If a previous PAN has indeed transferable devices then the web page seen in Figure 5(a) is presented. The devices from the previous PAN that the user wishes to transfer can be selected via a check

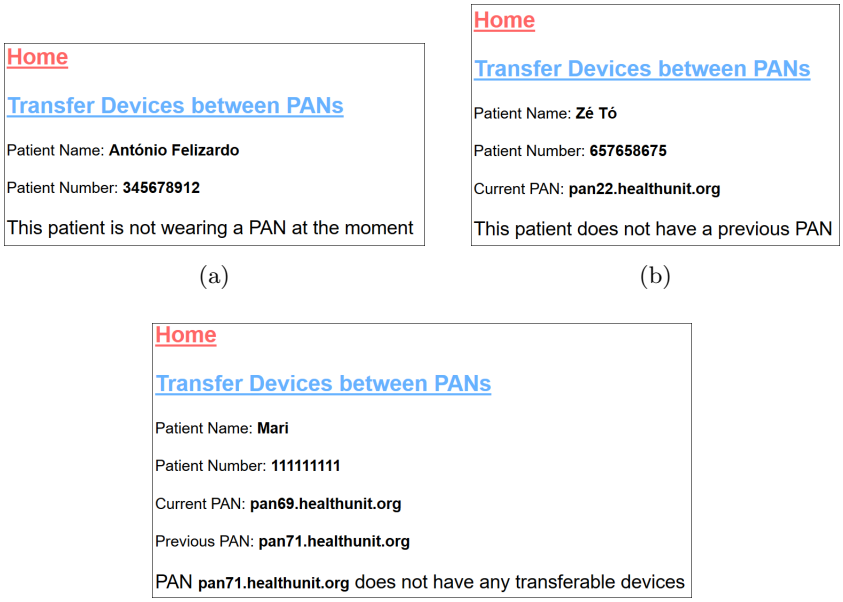


Figure 4: Unsuccessful transfers of devices between PANs - `transfer.php` (a), (b), (c).

box. The values read from this check box that have to be transmitted onto the next web page are the device serial number and manufacturer, because that is the composite primary key that identifies a certain device. For us it is also important to pass through the start time of connection of that device to the previous PAN, because that value is to be utilized in the query that updates the connection period equivalent to disconnecting the device from the previous PAN.

To pass all this information check boxes with an array are utilized. The check box value is composed of a concatenated string containing the serial number, the manufacturer and the start date, being all the fields separated with hash tags (#).

Other than the values coming from the check box, it is also important to pass through the domains of the previous PAN and the current PAN, which is done utilizing session variables.

After submitting the devices to be transferred, the web application advances to the web page - `execute_transfer.php`.

In this web page the queries that update

```
UPDATE Connects SET end = '$nowFormatted'
WHERE snum = '$snum[$i]'
  AND manuf = '$manuf[$i]'
  AND end = '2999-12-31 00:00:00
```

and insert the connections periods are performed.

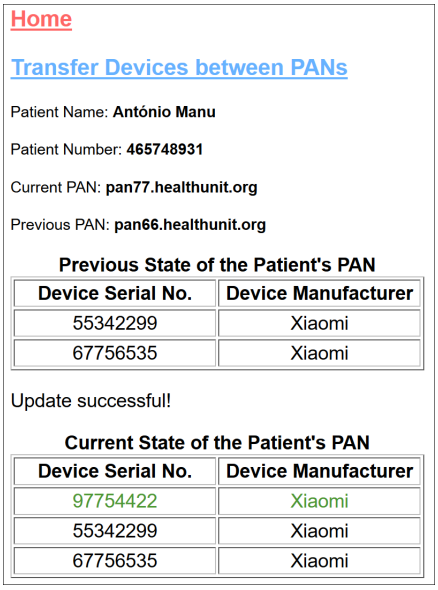
```
INSERT INTO Connects VALUES(
  '$nowplusoneFormatted',
    '2999-12-31 00:00:00',
    '$snum[$i]',
    '$manuf[$i]',
    '{$_SESSION['s_currentPAN']}')
```

The first query corresponds to updating the connection period of the device to the previous PAN from [start date, 2999-12-31 00:00:00] to [start date, now]. The second query corresponds to adding the connection period of the device to the current PAN as [now, 2999-12-31 00:00:00].

Before updating and inserting the values, it is verified that the desired periods are in table `Period`. If they are not they they have to be inserted, to prevent a foreign key constraint.

After a successful transfer, the web page presented in Figure 5(b) is presented, displaying how the current PAN was before the transfer, and how the current PAN is now, with the newly connected devices displaying in green color.

(a)



(b)

Figure 5: Successful transfer of devices between PANs - `execute_transfer.php` (a), (b).

5 Tables

The following tables were used in this project.

number	name	address
111111111	Mari	Da Street, n2, 4Esq, 1300-200 Oeiras
123456789	Roberto	Da Street, n3, 1Esq, 1000-999 Lisboa
234567891	João Lopes	Av. Rovisco Pais 1, 1000-900 Lisboa
345678912	António Felizardo	Estação Oriente, 1000-800 Lisboa
456789123	Miguel Prata	Rua da Prata, 1000-803 Lisboa
465748931	António Manu	Da Street, n2, 4Esq, 1300-200 Oeiras
567891234	Caldas	Alameda dos Oceanos, 1000-900 Lisboa
657658675	Zé Tó	Da Street, n2, 4Esq, 1300-200 Oeiras
678912345	Maria Joana	Rua dos Pastéis, 1000-500 Lisboa
987654321	Roberto	Da Street, n3, 1Esq, 1000-999 Lisboa
987654322	Toni	Da Street, n2, 4Esq, 1300-200 Oeiras

Figure 6: Patient

domain	phone
pan01.healthunit.org	212121212
pan02.healthunit.org	222121212
pan03.healthunit.org	234121212
pan04.healthunit.org	212340988
pan22.healthunit.org	775445324
pan33.healthunit.org	845763207
pan66.healthunit.org	666566754
pan69.healthunit.org	919191919
pan71.healthunit.org	911121122
pan77.healthunit.org	456948923
pan88.healthunit.org	676498532
pan99.healthunit.org	914457220

Figure 7: PAN

serialnum	manufacturer	description
11111111	Xiaomi	scale
12098765	Philips	scale
12345678	Philips	scale
12369901	Sony	rotating speed
12369902	Xiaomi	rotating speed
12369903	Philips	rotating speed
12369904	Philips	rotating speed
22222222	Xiaomi	scale
33333333	Xiaomi	scale
35647828	Philips	rotating speed
45545545	Samsung	rotating speed
55342299	Xiaomi	rotating speed
66666666	Philips	scale
67756535	Xiaomi	rotating speed
77777777	Matsui	rotating speed
87650943	Philips	blood pressure
97754422	Xiaomi	rotating speed
98765432	Philips	blood pressure

Figure 8: Device

snum	manuf	units
11111111	Xiaomi	kg
12098765	Philips	kg
12345678	Philips	kg
22222222	Xiaomi	kg
33333333	Xiaomi	kg
66666666	Philips	kg
87650943	Philips	mm Hg
98765432	Philips	mm Hg

Figure 9: Sensor

snum	manuf	units
12369901	Sony	rpm
35647828	Philips	rpm
45545545	Samsung	rpm
55342299	Xiaomi	rpm
67756535	Xiaomi	rpm
77777777	Matsui	rpm
97754422	Xiaomi	rpm

Figure 10: Actuator

nut4code	name
12345	Lisboa
24351	Oeiras
35374	Porto

Figure 11: Municipality

start	end
2000-07-02 08:00:00	2001-12-31 00:00:00
2012-03-02 08:00:00	2013-05-01 08:00:00
2013-07-02 08:00:00	2014-12-31 00:00:00
2013-07-02 08:00:00	2015-08-31 00:00:00
2013-07-02 08:00:00	2999-12-31 00:00:00
2014-01-01 08:00:00	2015-01-30 08:00:00
2014-05-29 08:56:21	2999-12-31 00:00:00
2014-12-04 08:00:00	2015-01-04 08:00:00
2014-12-04 08:00:00	2015-02-03 08:00:00
2015-01-01 08:00:00	2015-01-01 08:00:00
2015-01-01 08:00:00	2015-01-15 08:00:00
2015-01-01 08:00:00	2015-01-30 08:00:00
2015-01-01 08:00:00	2015-02-03 08:00:00
2015-01-02 08:00:00	2015-01-31 08:00:00
2015-01-02 08:00:00	2999-12-31 00:00:00
2015-01-03 08:00:00	2015-01-07 08:00:00
2015-01-03 08:00:00	2015-02-03 08:00:00
2015-01-12 00:00:00	2015-01-31 00:00:00
2015-01-12 00:00:00	2015-06-21 00:00:00
2015-01-12 00:00:00	2999-12-31 00:00:00
2015-02-01 00:00:00	2015-03-29 00:00:00
2015-02-01 08:00:00	2015-03-01 08:00:00
2015-02-02 08:00:00	2015-02-10 08:00:00
2015-03-13 00:00:00	2015-05-14 00:00:00
2015-03-20 08:00:00	2015-04-05 08:00:00
2015-04-13 00:00:00	2015-07-15 00:00:00
2015-05-02 08:00:00	2015-06-02 08:00:00
2015-06-02 08:00:00	2999-12-31 00:00:00
2015-06-03 08:00:00	2999-12-31 00:00:00
2015-06-14 00:00:00	2999-12-31 00:00:00
2015-06-15 00:00:00	2015-07-16 00:00:00
2015-06-20 00:00:00	2015-06-30 00:00:00
2015-06-22 00:00:00	2015-06-30 00:00:00
2015-07-02 08:00:00	2999-12-31 00:00:00
2015-07-14 00:00:00	2015-07-16 00:00:00
2015-07-31 00:00:00	2015-08-20 00:00:00
2015-08-10 00:00:00	2015-08-11 00:00:00
2015-08-12 00:00:00	2015-08-20 00:00:00
2015-08-21 00:00:00	2015-08-30 00:00:00
2015-08-25 00:00:00	2999-12-31 00:00:00
2015-08-26 00:00:00	2999-12-31 00:00:00
2015-09-15 00:00:00	2999-12-31 00:00:00
2015-09-15 08:05:52	2999-12-31 00:00:00
2015-09-17 00:00:00	2999-12-31 00:00:00
2015-09-18 08:05:52	2999-12-31 00:00:00
2015-11-01 12:30:24	2999-12-31 00:00:00
2015-12-13 17:50:44	2999-12-31 00:00:00
2017-01-14 00:00:00	2999-12-31 00:00:00
2017-01-15 00:00:00	2999-12-31 00:00:00
2017-01-16 00:00:00	2999-12-31 00:00:00
2017-07-15 08:00:00	2999-12-31 00:00:00
2017-08-16 09:00:55	2999-12-31 00:00:00
2017-08-17 08:05:52	2999-12-31 00:00:00
2017-08-30 08:00:15	2999-12-31 00:00:00

Figure 12: Period

snum	manuf	datetime	value
12098765	Philips	2015-01-30 08:00:15	35.00
66666666	Philips	2017-01-14 00:00:00	90.00
66666666	Philips	2017-01-15 00:00:00	90.00
66666666	Philips	2017-07-15 08:00:00	60.00
66666666	Philips	2017-08-30 08:00:15	80.00
87650943	Philips	2015-01-05 08:00:00	60.40
87650943	Philips	2015-02-05 08:00:00	60.40
87650943	Philips	2015-02-10 08:00:00	60.40
87650943	Philips	2015-03-20 08:00:00	60.40
87650943	Philips	2015-10-09 08:00:12	60.30
98765432	Philips	2015-10-01 08:00:15	60.00
98765432	Philips	2015-10-05 08:00:15	60.30
98765432	Philips	2015-10-09 08:00:15	60.10

Figure 13: Reading

snum	manuf	datetime	value
12369901	Sony	2015-09-15 08:05:52	30.43
12369901	Sony	2015-09-18 08:05:52	30.40
35647828	Philips	2015-10-09 09:00:55	50.45
77777777	Matsui	2017-08-16 09:00:55	50.45
77777777	Matsui	2017-08-17 08:05:52	30.40

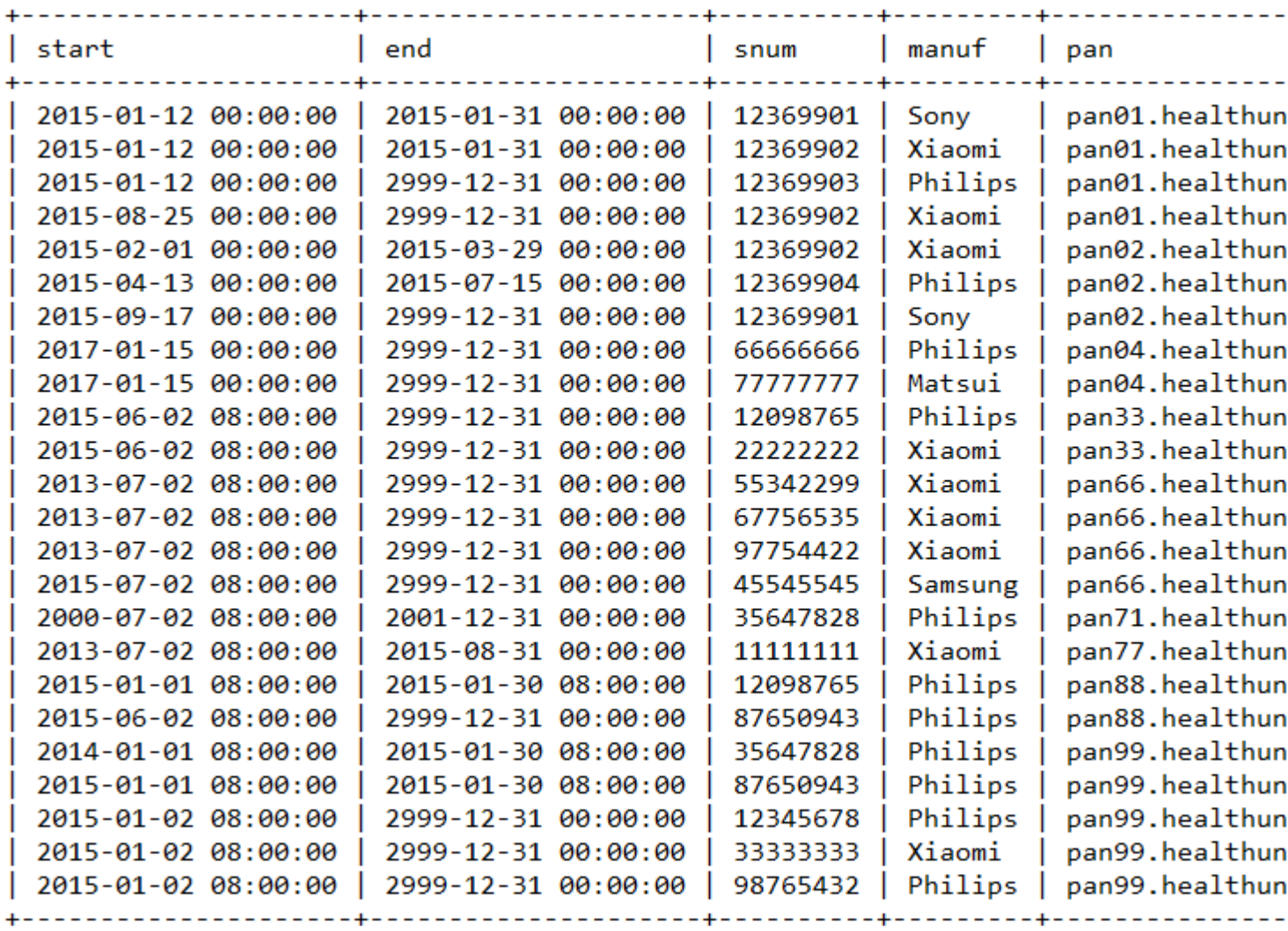
Figure 14: Setting

start	end	patient	pan
2015-01-12 00:00:00	2015-06-21 00:00:00	234567891	pan01.healthunit.org
2015-07-31 00:00:00	2015-08-20 00:00:00	567891234	pan01.healthunit.org
2015-08-21 00:00:00	2015-08-30 00:00:00	678912345	pan01.healthunit.org
2015-03-13 00:00:00	2015-05-14 00:00:00	345678912	pan02.healthunit.org
2015-06-15 00:00:00	2015-07-16 00:00:00	567891234	pan02.healthunit.org
2015-08-10 00:00:00	2015-08-11 00:00:00	678912345	pan02.healthunit.org
2015-09-15 00:00:00	2999-12-31 00:00:00	678912345	pan02.healthunit.org
2015-06-22 00:00:00	2015-06-30 00:00:00	234567891	pan03.healthunit.org
2017-01-16 00:00:00	2999-12-31 00:00:00	567891234	pan04.healthunit.org
2015-07-02 08:00:00	2999-12-31 00:00:00	657658675	pan22.healthunit.org
2015-02-01 08:00:00	2015-03-01 08:00:00	123456789	pan33.healthunit.org
2015-07-02 08:00:00	2999-12-31 00:00:00	987654322	pan33.healthunit.org
2013-07-02 08:00:00	2014-12-31 00:00:00	465748931	pan66.healthunit.org
2015-07-02 08:00:00	2999-12-31 00:00:00	111111111	pan69.healthunit.org
2000-07-02 08:00:00	2001-12-31 00:00:00	111111111	pan71.healthunit.org
2015-07-02 08:00:00	2999-12-31 00:00:00	465748931	pan77.healthunit.org
2015-01-01 08:00:00	2015-01-30 08:00:00	987654321	pan88.healthunit.org
2015-06-03 08:00:00	2999-12-31 00:00:00	987654321	pan88.healthunit.org
2000-07-02 08:00:00	2001-12-31 00:00:00	465748931	pan99.healthunit.org
2015-01-01 08:00:00	2015-01-30 08:00:00	123456789	pan99.healthunit.org
2015-06-03 08:00:00	2999-12-31 00:00:00	123456789	pan99.healthunit.org

Figure 15: Wears

start	end	patient	muni
2015-01-02 08:00:00	2999-12-31 00:00:00	123456789	12345
2015-01-02 08:00:00	2999-12-31 00:00:00	987654321	12345
2015-01-02 08:00:00	2999-12-31 00:00:00	987654322	24351
2015-01-02 08:00:00	2999-12-31 00:00:00	465748931	35374

Figure 16: Lives



start	end	snum	manuf	pan
2015-01-12 00:00:00	2015-01-31 00:00:00	12369901	Sony	pan01.healthun
2015-01-12 00:00:00	2015-01-31 00:00:00	12369902	Xiaomi	pan01.healthun
2015-01-12 00:00:00	2999-12-31 00:00:00	12369903	Philips	pan01.healthun
2015-08-25 00:00:00	2999-12-31 00:00:00	12369902	Xiaomi	pan01.healthun
2015-02-01 00:00:00	2015-03-29 00:00:00	12369902	Xiaomi	pan02.healthun
2015-04-13 00:00:00	2015-07-15 00:00:00	12369904	Philips	pan02.healthun
2015-09-17 00:00:00	2999-12-31 00:00:00	12369901	Sony	pan02.healthun
2017-01-15 00:00:00	2999-12-31 00:00:00	66666666	Philips	pan04.healthun
2017-01-15 00:00:00	2999-12-31 00:00:00	77777777	Matsui	pan04.healthun
2015-06-02 08:00:00	2999-12-31 00:00:00	12098765	Philips	pan33.healthun
2015-06-02 08:00:00	2999-12-31 00:00:00	22222222	Xiaomi	pan33.healthun
2013-07-02 08:00:00	2999-12-31 00:00:00	55342299	Xiaomi	pan66.healthun
2013-07-02 08:00:00	2999-12-31 00:00:00	67756535	Xiaomi	pan66.healthun
2013-07-02 08:00:00	2999-12-31 00:00:00	97754422	Xiaomi	pan66.healthun
2015-07-02 08:00:00	2999-12-31 00:00:00	45545545	Samsung	pan66.healthun
2000-07-02 08:00:00	2001-12-31 00:00:00	35647828	Philips	pan71.healthun
2013-07-02 08:00:00	2015-08-31 00:00:00	11111111	Xiaomi	pan77.healthun
2015-01-01 08:00:00	2015-01-30 08:00:00	12098765	Philips	pan88.healthun
2015-06-02 08:00:00	2999-12-31 00:00:00	87650943	Philips	pan88.healthun
2014-01-01 08:00:00	2015-01-30 08:00:00	35647828	Philips	pan99.healthun
2015-01-01 08:00:00	2015-01-30 08:00:00	87650943	Philips	pan99.healthun
2015-01-02 08:00:00	2999-12-31 00:00:00	12345678	Philips	pan99.healthun
2015-01-02 08:00:00	2999-12-31 00:00:00	33333333	Xiaomi	pan99.healthun
2015-01-02 08:00:00	2999-12-31 00:00:00	98765432	Philips	pan99.healthun

Figure 17: Connects

6 Source Code

```
<html>
  <head>
    <title>SIBD Project - Group 16</title>
  </head>
  <body link="#66b2ff">
    <font face="Helvetica">
    <h1><center><font color="#ff6666">Welcome to our Health Care Center</font></
    center></h1>
    <hr/>
    <h2><center><a href="patient_records.php">Access Patient Records</a></center></h2
    >
    <h2><center><a href="transfer_devices.php">Transfer Devices between PANs</a></
    center></h2>
    </font>
  </body>
</hmtl>
```

```
<?php session_start(); ?>
<html>
  <head>
    <title>Transfer Devices</title>
  </head>
  <body link="#ff6666">
    <font face="Helvetica">
    <h3><a href="index.html">Home</a></h3>
    <h3><a href="transfer_devices.php"><font color="#66b2ff">Transfer Devices between
     PANs</font></a></h3>
<?php
  ini_set('display_errors', 'On');
  error_reporting(E_ALL);

  $host = "db.ist.utl.pt";
  $user = "ist173099";
  $pass = "mile6613";
  $dsn = "mysql:host=$host;dbname=$user";
  try {
    $connection = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception) {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }

  $snum = array();
```

```
$manuf = array();
$start = array();

foreach ($_REQUEST as $info => $value) { // $_REQUEST is an associative array (key => value)
  if ($info == 'device_info') {
    foreach($_REQUEST[$info] as $device_info) {
      $device_info_pieces = explode("#", $device_info);
      $snum[] = $device_info_pieces[0]; // snum
      $manuf[] = $device_info_pieces[1]; // manuf
      $start[] = $device_info_pieces[2]; // start
    }
  }
  else {
    $info = $value;
  }
}

echo("<p><font size=\"2\">Patient Name: <strong>"."{$_SESSION['s_pname']}"."</strong></font></p>");
echo("<p><font size=\"2\">Patient Number: <strong>"."{$_SESSION['s_pnum']}"."</strong></font></p>");
echo("<p><font size=\"2\">Current PAN: <strong>"."{$_SESSION['s_currentPAN']}"."</strong></font></p>");
echo("<p><font size=\"2\">Previous PAN: <strong>"."{$_SESSION['s_previousPAN']}"."</strong></font></p>");

if(empty($snum)) {
  echo("No transferable devices were selected");
}

else {
  $sql = "CREATE TABLE tableBeforePAN(snum numeric(8,0), manuf varchar(255))";
  $result = $connection->query($sql);
  if ($result == FALSE) {
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
  }

  $sql = "INSERT INTO tableBeforePAN (snum, manuf)
          SELECT Connects.snum, Connects.manuf
          FROM Connects
          WHERE Connects.end = '2999-12-31 00:00:00'
            AND Connects.pan = '{$_SESSION['s_currentPAN']}'";
  $result = $connection->query($sql);
  if ($result == FALSE) {
```

```
71      $info = $connection->errorInfo();
72      echo("<p>Error: {$info[2]}</p>");
73      exit();
74    }
75
76    $sql = "SELECT * FROM tableBeforePAN";
77    $result = $connection->query($sql);
78    if ($result == FALSE) {
79      $info = $connection->errorInfo();
80      echo("<p>Error: {$info[2]}</p>");
81      exit();
82    }
83
84    echo("<table border=\"1\">");
85    echo("<caption><strong>Previous State of the Patient's PAN</strong></caption>");
86
87    if($result->rowCount() == 0) {
88        echo("<col width=\"400\">");
89        echo("<tr><td align=\"center\">");
90        echo("No devices connected to this PAN before transfer");
91        echo("</td></tr>");
92    }
93
94    else {
95        echo("<col width=\"170\"><col width=\"170\">");
96        echo("<tr><th>Device Serial No.</th><th>Device Manufacturer</th></tr>");
97        foreach($result as $row) {
98          echo("<tr><td align=\"center\">");
99          echo($row['snum']);
100         echo("</td><td align=\"center\">");
101         echo($row['manuf']);
102         echo("</td></tr>");
103       }
104   }
105
106   echo("</table>");
107
108   for ($i = 0; $i < count($snum); $i++) {
109     $now = new DateTime();
110     $nowFormatted = $now->format('Y-m-d H:i:s');
111     $sql = "SELECT Period.start, Period.end
112             FROM Period
113             WHERE Period.start = '$start[$i]'
114               AND Period.end = '$nowFormatted'"; // verificar se o periodo ja
    existe na tabela, se sim entao saltar o insert
115     $result = $connection->query($sql);
116     if ($result == FALSE) {
```

```
117         $info = $connection->errorInfo();
118         echo("<p>Error: {$info[2]}</p>");
119         exit();
120       }
121
122       if($result->rowCount() == 0) { // significa que o periodo nao existe e tem de
    ser inserido na tabela
123         $sql = "INSERT INTO Period VALUES ('$start[$i]', '$nowFormatted')"; // para
    desligar o device da previous PAN
124         $result = $connection->query($sql);
125         if ($result == FALSE) {
126           $info = $connection->errorInfo();
127           echo("<p>Error: {$info[2]}</p>");
128           exit();
129         }
130       }
131
132       $sql = "UPDATE Connects SET end = '$nowFormatted'
133               WHERE snum = '$snum[$i]'
134                 AND manuf = '$manuf[$i]'
135                 AND end = '2999-12-31 00:00:00'";
136       $result = $connection->query($sql);
137       if ($result == FALSE) {
138         $info = $connection->errorInfo();
139         echo("<p>Error: {$info[2]}</p>");
140         exit();
141       }
142
143       $nowplusone = $now->add(new DateInterval('PT1S'));
144       $nowplusoneFormatted = $nowplusone->format('Y-m-d H:i:s');
145       $sql = "SELECT Period.start, Period.end
146               FROM Period
147               WHERE Period.start = '$nowplusoneFormatted'
148                 AND Period.end = '2999-12-31 00:00:00'"; // verificar se o periodo ja
     existe na tabela, se sim entao saltar o insert
149       $result = $connection->query($sql);
150       if ($result == FALSE) {
151         $info = $connection->errorInfo();
152         echo("<p>Error: {$info[2]}</p>");
153         exit();
154       }
155
156       if($result->rowCount() == 0) { // significa que o periodo nao existe e tem de
    ser inserido na tabela
157         $sql = "INSERT INTO Period VALUES ('$nowplusoneFormatted', '2999-12-31
    00:00:00')"; // para ligar o device a current PAN
158         $result = $connection->query($sql);
```

```
159         if ($result == FALSE) {
160           $info = $connection->errorInfo();
161           echo("<p>Error: {$info[2]}</p>");
162           exit();
163         }
164       }
165
166       $sql = "INSERT INTO Connects VALUES('$nowplusoneFormatted', '2999-12-31
    00:00:00', '$snum[$i]', '$manuf[$i]', '{$_SESSION['s_currentPAN']}')";
167       $result = $connection->query($sql);
168       if ($result == FALSE) {
169         $info = $connection->errorInfo();
170         echo("<p>Error: {$info[2]}</p>");
171         echo("Error in transfering devices from PAN ");
172         echo("<font size=\"2\"><strong>"."{$_SESSION['s_previousPAN']}"."</strong></
    font>");
173         echo(" to PAN ");
174         echo("<font size=\"2\"><strong>"."{$_SESSION['s_currentPAN']}"."</strong></
    font>");
175         echo("<p></p>");
176         exit();
177       }
178     }
179     echo("<p>Update successful!</p>");
180
181     $sql = "CREATE TABLE tableAfterPAN(snum numeric(8,0), manuf varchar(255))";
182     $result = $connection->query($sql);
183     if ($result == FALSE) {
184       $info = $connection->errorInfo();
185       echo("<p>Error: {$info[2]}</p>");
186       exit();
187     }
188
189     $sql = "INSERT INTO tableAfterPAN (snum, manuf)
190             SELECT Connects.snum, Connects.manuf
191             FROM Connects
192             WHERE Connects.end = '2999-12-31 00:00:00'
193               AND Connects.pan = '{$_SESSION['s_currentPAN']}'";
194
195     $result = $connection->query($sql);
196     if ($result == FALSE) {
197       $info = $connection->errorInfo();
198       echo("<p>Error: {$info[2]}</p>");
199       exit();
200     }
201
202     $sql = "SELECT * FROM tableAfterPAN";
```

```
203     $result = $connection->query($sql);
204     if ($result == FALSE) {
205       $info = $connection->errorInfo();
206       echo("<p>Error: {$info[2]}</p>");
207       exit();
208     }
209 
210     $sql = "SELECT DISTINCT snum, manuf
211             FROM tableAfterPAN
212             WHERE (snum, manuf) NOT IN (SELECT DISTINCT snum, manuf from
    tableBeforePAN)"; // devices que foram inseridos
213     $result = $connection->query($sql);
214     if ($result == FALSE) {
215       $info = $connection->errorInfo();
216       echo("<p>Error: {$info[2]}</p>");
217       exit();
218     }
219 
220     echo("<table border=\"1\">");
221     echo("<caption><strong>Current State of the Patient's PAN</caption>");
222 
223     echo("<col width=\"170\"><col width=\"170\">");
224     echo("<tr><th>Device Serial No.</th><th>Device Manufacturer</th></tr>");
225     foreach($result as $row) {
226       echo("<tr><td align=\"center\"><font color=\"#4d9933\">");
227       echo($row['snum']);
228       echo("</font></td><td align=\"center\"><font color=\"#4d9933\">");
229       echo($row['manuf']);
230       echo("</font></td></tr>");
231     }
232 
233     $sql = "SELECT * FROM tableBeforePAN"; // devices que ja estavam na PAN antes da
    transferencia
234     $result = $connection->query($sql);
235     if ($result == FALSE) {
236       $info = $connection->errorInfo();
237       echo("<p>Error: {$info[2]}</p>");
238       exit();
239     }
240 
241     foreach($result as $row) {
242       echo("<tr><td align=\"center\">");
243       echo($row['snum']);
244       echo("</td><td align=\"center\">");
245       echo($row['manuf']);
246       echo("</td></tr>");
247     }
```

```
248
249     echo("</table>");
250
251     $sql = "DROP TABLE IF EXISTS tableBeforePAN";
252     $result = $connection->query($sql);
253     if ($result == FALSE) {
254       $info = $connection->errorInfo();
255       echo("<p>Error: {$info[2]}</p>");
256       exit();
257     }
258
259     $sql = "DROP TABLE IF EXISTS tableAfterPAN";
260     $result = $connection->query($sql);
261     if ($result == FALSE) {
262       $info = $connection->errorInfo();
263       echo("<p>Error: {$info[2]}</p>");
264       exit();
265     }
266   }
267
268   session_destroy();
269
270   $connection = null;
271 ?>
272     </font>
273   </body>
274 </html>
```

```
1  <html>
2    <head>
3      <title>Patient Records</title>
4    </head>
5    <body link="#ff6666">
6      <font face="Helvetica">
7      <form action="select_patient_records.php" method="post">
8        <h3><a href="index.html">Home</a></h3>
9        <h3><font color="#66b2ff">Acess Patient Records</font></h3>
10       <p>Patient Name:
11         <input type="text" name="name"/>
12         <input type="submit" value="Submit"/>
13       </p>
14       <p><font size="2">or just press "submit" to display all registered patients</
       font></p>
15     </form>
16     </font>
17   </body>
18 </hmtl>
```

```
<html>
  <head>
    <title>Patient Records</title>
  </head>
  <body link="#ff6666">
    <font face="Helvetica">
    <h3><a href="index.html">Home</a></h3>
    <h3><a href="patient_records.php"><font color="#66b2ff">Acess Patient Records</
   font></a></h3>
<?php
  $host = "db.ist.utl.pt";
  $user = "ist173099";
  $pass = "mile6613";
  $dsn = "mysql:host=$host;dbname=$user";
  try {
    $connection = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception) {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }

  $name = $_REQUEST['name'];
  $number = $_REQUEST['number'];

  $sql = "SELECT DISTINCT Reading.snum, Reading.manuf, Reading.value, Sensor.units,
    Reading.datetime
          FROM Patient, Wears, Connects, Reading, Sensor
          WHERE Wears.patient = '$number'
            AND Wears.pan = Connects.pan
            AND Connects.snum = Reading.snum
            AND Connects.manuf = Reading.manuf
            AND Sensor.snum = Reading.snum
            AND Sensor.manuf = Reading.manuf
            AND Reading.datetime BETWEEN Connects.start AND Connects.end
            AND Reading.datetime BETWEEN Wears.start AND Wears.end";

  $result = $connection->query($sql);
  if ($result == FALSE) {
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
  }

  echo("<p><font size=\"2\">Patient Name: <strong>$name</strong></p>");
```

```
46  echo("<p>Patient Number: <strong>$number</strong></font></p>");
47
48  echo("<table border=\"1\">");
49  echo("<caption><strong>Readings</strong></caption>");
50
51  if($result->rowCount() == 0) {
52    echo("<col width=\"300\">");
53    echo("<tr><td align=\"center\">");
54    echo("No readings found for this patient");
55    echo("</td></tr>");
56  }
57
58  else {
59    echo("<col width=\"170\"><col width=\"170\"><col width=\"170\"><col width
      =\"170\"><col width=\"170\">");
60    echo("<tr><th>Device Serial No.</th><th>Device Manufacturer</th><th>Value</th><th
      >Units</th><th>Date and Time</th></tr>");
61    foreach($result as $row) {
62      echo("<tr><td align=\"center\">");
63      echo($row['snum']);
64      echo("</td><td align=\"center\">");
65      echo($row['manuf']);
66      echo("</td><td align=\"center\">");
67      echo($row['value']);
68      echo("</td><td align=\"center\">");
69      echo($row['units']);
70      echo("</td><td align=\"center\">");
71      echo($row['datetime']);
72      echo("</td></tr>");
73    }
74  }
75
76  echo("</table>");
77
78  echo("<p></p>");
79
80  $sql = "SELECT DISTINCT Setting.snum, Setting.manuf, Setting.value, Actuator.units,
       Setting.datetime
81          FROM Patient, Wears, Connects, Setting, Actuator
82          WHERE Wears.patient = '$number'
83            AND Wears.pan = Connects.pan
84            AND Connects.snum = Setting.snum
85            AND Connects.manuf = Setting.manuf
86            AND Actuator.snum = Setting.snum
87            AND Actuator.manuf = Setting.manuf
88            AND Setting.datetime BETWEEN Connects.start AND Connects.end
89            AND Setting.datetime BETWEEN Wears.start AND Wears.end";
```

```
90
91    $result = $connection->query($sql);
92    if ($result == FALSE) {
93      $info = $connection->errorInfo();
94      echo("<p>Error: {$info[2]}</p>");
95      exit();
96    }
97
98    echo("<table border=\"1\">");
99    echo("<caption><strong>Settings</strong></caption>");
100
101   if($result->rowCount() == 0) {
102     echo("<col width=\"300\">");
103     echo("<tr><td align=\"center\">");
104     echo("No settings found for this patient");
105     echo("</td></tr>");
106   }
107
108   else {
109     echo("<col width=\"170\"><col width=\"170\"><col width=\"170\"><col width
      =\"170\"><col width=\"170\">");
110     echo("<tr><th>Device Serial No.</th><th>Device Manufacturer</th><th>Value</th><th
      >Units</th><th>Date and Time</th></tr>");
111     foreach($result as $row) {
112       echo("<tr><td align=\"center\">");
113       echo($row['snum']);
114       echo("</td><td align=\"center\">");
115       echo($row['manuf']);
116       echo("</td><td align=\"center\">");
117       echo($row['value']);
118       echo("</td><td align=\"center\">");
119       echo($row['units']);
120       echo("</td><td align=\"center\">");
121       echo($row['datetime']);
122       echo("</td></tr>");
123     }
124   }
125
126   echo("</table>");
127
128   $connection = null;
129 ?>
130     </font>
131   </body>
132 </html>
```

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

```
    <title>Patient Records</title>
  </head>
  <body>
    <font face="Helvetica">
    <h3><a href="index.html"><font color="#ff6666">Home</font></a></h3>
    <h3><a href="patient_records.php"><font color="#66b2ff">Acess Patient Records</
    font></a></h3>
<?php
  $host = "db.ist.utl.pt";
  $user = "ist173099";
  $pass = "mile6613";
  $dsn = "mysql:host=$host;dbname=$user";
  try {
    $connection = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception) {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }

  $name = $_REQUEST['name'];

  $sql = "SELECT Patient.name, Patient.number
          FROM Patient
          WHERE Patient.name like '%$name%'
          ORDER BY Patient.name";

  $result = $connection->query($sql);
  if ($result == FALSE) {
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
  }

  if($result->rowCount() == 0) {
    echo("No patients named <strong>$name</strong> were found");
  }

  else {
    echo("Displaying results for patients named <strong>$name</strong>:");
    echo("<p></p>");
    echo("<table border=\"0\" cellspacing=\"10\">\n");
    echo("<col width=\"170\"><col width=\"100\"><col width=\"120\">");
    echo("<tr><th>Name</th><th>Number</th></tr>");
    foreach($result as $row) {
```

```
      echo("<tr>\n");
      echo("<td align=\"center\">{$row['name']}</td>\n");
      echo("<td align=\"center\">{$row['number']}</td>\n");
      echo("<td align=\"center\"><a href=\"records.php?number=");
      echo($row['number']);
      echo("&name=");
      echo($row['name']);
      echo("\"><font color=\"#007fff\">Show records</font></a></td>\n");
      echo("</tr>\n");
    }
    echo("</table>");
  }

  $connection = null;
?>
    </font>
  </body>
</html>
```

```
<html>
  <head>
    <title>Transfer Devices</title>
  </head>
  <body>
    <font face="Helvetica">
    <h3><a href="index.html"><font color="#ff6666">Home</font></a></h3>
    <h3><a href="transfer_devices.php"><font color="#66b2ff">Transfer Devices between
     PANs</font></a></h3>
<?php
  $host = "db.ist.utl.pt";
  $user = "ist173099";
  $pass = "mile6613";
  $dsn = "mysql:host=$host;dbname=$user";
  try {
    $connection = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception) {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }

  $name = $_REQUEST['name'];

  $sql = "SELECT Patient.name, Patient.number
          FROM Patient
          WHERE Patient.name like '%$name%'
```

```
          ORDER BY Patient.name";

  $result = $connection->query($sql);
  if ($result == FALSE) {
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
  }

  if($result->rowCount() == 0) {
    echo("No patients named <strong>$name</strong> were found");
  }

  else {
    echo("Displaying results for patients named <strong>$name</strong>:");
    echo("<p></p>");
    echo("<table border=\"0\" cellspacing=\"10\">\n");
    echo("<col width=\"170\"><col width=\"100\"><col width=\"150\">");
    echo("<tr><th>Name</th><th>Number</th></tr>");
    foreach($result as $row) {
      echo("<tr>\n");
      echo("<td align=\"center\">{$row['name']}</td>\n");
      echo("<td align=\"center\">{$row['number']}</td>\n");
      echo("<td align=\"center\"><a href=\"transfer.php?number=");
      echo($row['number']);
      echo("&name=");
      echo($row['name']);
      echo("\"><font color=\"#007fff\">Transfer devices</font></a></td>\n");
      echo("</tr>\n");
    }
    echo("</table>");
  }

  $connection = null;
?>
    </font>
  </body>
</html>
```

At the beggining of the next code file the following instruction is needed (the report wouldn't compile): $<?php\ session\_start();?>$

```
<html>
  <head>
    <title>Transfer Devices</title>
  </head>
  <body link="#ff6666">
    <font face="Helvetica">
```

```
    <form action="execute_transfer.php" method="post">
    <h3><a href="index.html">Home</a></h3>
    <h3><a href="transfer_devices.php"><font color="#66b2ff">Transfer Devices between
     PANs</font></a></h3>
<?php
  $host = "db.ist.utl.pt";
  $user = "ist173099";
  $pass = "mile6613";
  $dsn = "mysql:host=$host;dbname=$user";
  try {
    $connection = new PDO($dsn, $user, $pass);
  }
  catch(PDOException $exception) {
    echo("<p>Error: ");
    echo($exception->getMessage());
    echo("</p>");
    exit();
  }

  $name = $_REQUEST['name'];
  $number = $_REQUEST['number'];

  $sql = "SELECT Wears.pan
          FROM Wears
          WHERE Wears.patient = '$number'
            AND Wears.end = '2999-12-31 00:00:00'";

  $result = $connection->query($sql);
  if ($result == FALSE) {
    $info = $connection->errorInfo();
    echo("<p>Error: {$info[2]}</p>");
    exit();
  }

  echo("<p><font size=\"2\">Patient Name: <strong>$name</strong></p>");
  echo("<p>Patient Number: <strong>$number</strong></font></p>");

  if($result->rowCount() == 0) {
    echo("This patient is not wearing a PAN at the moment");
  }

  else {
    $currentPAN = $result->fetchColumn();
    echo("<p><font size=\"2\">Current PAN: <strong>$currentPAN</strong></font></p>");

    $sql = "SELECT Wears.pan, Wears.end
            FROM Wears
```

```
53            WHERE Wears.patient = '$number'
54            ORDER BY Wears.end DESC
55            LIMIT 1,1"; // para aceder a segunda linha da tabela de resultados que
    contem a penultima PAN
56
57     $result = $connection->query($sql);
58     if ($result == FALSE) {
59       $info = $connection->errorInfo();
60       echo("<p>Error: {$info[2]}</p>");
61       exit();
62     }
63
64     if($result->rowCount() == 0) {
65       echo("This patient does not have a previous PAN");
66     }
67
68     else {
69       $previousPAN_data = $result->fetch(PDO::FETCH_ASSOC); // indexed by column name
70       $previousPAN = $previousPAN_data["pan"];
71       $previousPAN_end = $previousPAN_data["end"];
72
73       echo("<p><font size=\"2\">Previous PAN: <strong>$previousPAN</strong></font></p
    >");
74
75       if($previousPAN === $currentPAN) { //the two variables are of the same type
76         echo("The current PAN is the same as the previous PAN");
77       }
78
79       else {
80         $sql = "SELECT Connects.snum, Connects.manuf, Connects.start
81                 FROM Connects
82                 WHERE Connects.pan = '$previousPAN'
83                   AND Connects.start < '$previousPAN_end'
84                   AND Connects.end = '2999-12-31 00:00:00'";
85                   // garantir que devices foram ligados a previous PAN durante o
    tempo que o paciente esteve ligado a ela
86                 // garantir que os devices ainda estao ligados a previous PAN
87                 // preciso do start para depois poder inserir na tabela Period[
    start, now] para desligar da previous PAN
88
89         $result = $connection->query($sql);
90         if ($result == FALSE) {
91           $info = $connection->errorInfo();
92           echo("<p>Error: {$info[2]}</p>");
93           exit();
94         }
95
```

```
96          if($result->rowCount() == 0) {
97            echo("PAN <font size=\"2\"><strong>$previousPAN</strong></font> does not
    have any transferable devices");
98          }
99
100         else {
101           echo("Displaying devices than can be transfered from PAN ");
102           echo("<font size=\"2\"><strong>$previousPAN</strong></font>");
103           echo(" to PAN ");
104           echo("<font size=\"2\"><strong>$currentPAN</strong></font>");
105           echo("<p></p>");
106
107           foreach($result as $row) {
108             echo("<input type=\"checkbox\" name=\"device_info[]\" value=\"{$row['snum
    ']}#{$row['manuf']}#{$row['start']}\"/>
109               <font size=\"2.5\"><strong>Serial Number</strong></font>: {$row['snum
    ']}
110               / <font size=\"2.5\"><strong>Manufacturer</strong></font>: {$row['manuf
    ']}<br/>");
111           }
112
113           echo("<p><input type=\"submit\" value=\"Submit\"/></p>");
114         }
115       }
116     }
117   }
118
119   $_SESSION['s_pname'] = $name;
120   $_SESSION['s_pnum'] = $number;
121   $_SESSION['s_previousPAN'] = $previousPAN;
122   $_SESSION['s_currentPAN'] = $currentPAN;
123
124   $connection = null;
125 ?>
126     </form>
127     </font>
128   </body>
129 </html>
```

```
1  <html>
2    <head>
3      <title>Transfer Devices</title>
4    </head>
5    <body link="#ff6666">
6      <font face="Helvetica">
7      <form action="select_patient_transfer.php" method="post">
8        <h3><a href="index.html">Home</a></h3>
9        <h3><font color="#66b2ff">Transfer Devices between PANs</font></h3>
```

```
10      <p>Patient Name:
11        <input type="text" name="name"/>
12        <input type="submit" value="Submit"/>
13      </p>
14      <p><font size="2">or just press "submit" to display all registered patients</
    font></p>
15     </form>
16     </font>
17   </body>
18 </hmtl>
```