VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

DATABASE SYSTEMS (LAB) (C02014)

Assignment 2

QUARANTINE CAMP DATABASE

Instructor:	Mr. Phan Trong Nhan	
Class:	CC05	
Students:	Le Tien Loc	1952328
	Tran Tien Phat	1952386
	Nguyen Binh Minh	1952846
	Tang The Huy	2053053
	Huynh Thi Anh Thu	1952131
	Nakamura Ryuta	1752004

HO CHI MINH CITY, NOVEMBER 2021

Contents

1 Topic

2 Requirements

2.1 Physical Database Design

2.1.1 Implementing the database

```
CREATE TABLE PERSON (
    Person_ID           CHAR(6)         PRIMARY KEY,
    Person_address      VARCHAR(50),
    Person_name         VARCHAR(30)     NOT NULL,
    Gender              CHAR(1),
    Phone_no            CHAR(10),
    CONSTRAINT          check_person_gender CHECK (Gender = 'M' or Gender =
    'F' or Gender IS NULL)
);

CREATE TABLE NURSE (
    Nurse_ID            CHAR(6)         PRIMARY KEY,
    CONSTRAINT          fk_nurse_person_id FOREIGN KEY (Nurse_ID)
                            REFERENCES PERSON(Person_ID)
                            ON DELETE CASCADE
);

CREATE TABLE DOCTOR (
    Doctor_ID           CHAR(6)         PRIMARY KEY,
    CONSTRAINT          fk_doctor_person_id FOREIGN KEY (Doctor_ID)
                            REFERENCES PERSON(Person_ID)
                            ON DELETE CASCADE
);

CREATE TABLE STAFF (
    Staff_ID            CHAR(6)         PRIMARY KEY,
    CONSTRAINT          fk_staff_person_id FOREIGN KEY (Staff_ID)
                           REFERENCES PERSON(Person_ID)
                           ON DELETE CASCADE
);

CREATE TABLE VOLUNTEER (
    Volunteer_ID        CHAR(6)         PRIMARY KEY,
    CONSTRAINT          fk_volunteer_person_id FOREIGN KEY (Volunteer_ID)
                           REFERENCES PERSON(Person_ID)
                           ON DELETE CASCADE
);

CREATE TABLE MANAGER (
    Manager_ID          CHAR(6)         PRIMARY KEY,
```

```
    Doctor_ID           CHAR(6),
    CONSTRAINT          fk_manager_person_id FOREIGN KEY (Manager_ID)
                                REFERENCES PERSON(Person_ID)
                                ON DELETE CASCADE,
    CONSTRAINT          fk_manager_doctor_id FOREIGN KEY (Doctor_ID)
                                REFERENCES DOCTOR(Doctor_ID)
                                ON DELETE SET NULL
);
```

- TABLE PERSON
  - **Person_ID**: is a fixed-length string with 6 characters and set to PRIMARY KEY. For example: NUR001, DOC001, etc.
  - **Person_address**: is a variable-length string with maximum 50 characters.
  - **Person_name**: is a variable-length string with maximum 30 characters and is a NOT NULL attribute. Since the person name is the name of the staff and it helps them and the patient communicate with each other easily.
  - **Gender**: is indicated by 1-character string.
  - **Phone_no**: is a fixed-length string with 10 characters.
  - **CONSTRAINT check_person_gender**: check whether the specified value for Gender attribute is correct. 'F' for female and 'M' for male.
- TABLES NURSE, DOCTOR, STAFF and VOLUNTEER
  - **Nurse_ID**, **Doctor_ID**, **Staff_ID** and **Volunteer_ID**: are a fixed-length strings with 6 characters and set to PRIMARY KEY.
  - **CONSTRAINTS fk_nurse_person_id**, **fk_doctor_person_id**, **fk_staff_person_id** and **fk_volunteer_person_id**: are FOREIGN KEY referencing to Person_ID of PERSON table.
    ON DELETE CASCADE constraint is used to delete the rows from the NURSE, DOCTOR, STAFF and VOLUNTEER tables automatically, when the corresponding rows from the PERSON table are deleted.
- TABLE MANAGER
  - **Manager_ID**: is a fixed-length string with 6 characters and set to PRIMARY KEY.
  - **Doctor_ID**: is a fixed-length string with 6 characters.
  - **CONSTRAINTS fk_manager_person_id**: is FOREIGN KEY referencing to Person_ID of PERSON table.
  - **CONSTRAINTS fk_doctor_person_id**: is FOREIGN KEY referencing to Doctor_ID of DOCTOR table for the relationship *"One doctor will be designated as the head of the camp"*.

```
CREATE TABLE MEDICATION (
    Unique_code         CHAR(7)         PRIMARY KEY,
    Med_name            VARCHAR(30)     UNIQUE,
    Effect              VARCHAR(100),
    Price               INT,
```

```
    Expiration_Date     DATE
);
```

- TABLE MEDICATION
  - **Unique_code**: is a fixed-length string with 7 characters and set to PRIMARY KEY. For example: MED0001, MED0002...
  - **Med_name**: is a variable-length string with maximum 30 characters and set to UNIQUE to ensures that all medication name are different.
  - **Effect**: is a variable-length string with maximum 100 characters, indicating the effects of the medication.
  - **Price**: is a integer number, indicating the price unit of the medication.
  - **Expiration_Date**: is a DATE type attriute, it means the expiry date of the medicine.

```
CREATE TABLE PATIENT (
    Patient_no          CHAR(6)         PRIMARY KEY,
    Patient_name        VARCHAR(30)     NOT NULL,
    ID_no               CHAR(9)         NOT NULL,
    Phone_no            CHAR(10),
    Gender              CHAR(1),
    Patient_address     VARCHAR(50),
    Discharge_date      DATE,
    Nurse_ID            CHAR(6)         NOT NULL,
    Admission_date      DATE            NOT NULL,
    Location_before     VARCHAR(50),
    Staff_ID            CHAR(6)         NOT NULL,
    CONSTRAINT          Check_Gender CHECK (Gender = 'M' or Gender = 'F' or
    Gender IS NULL)
    CONSTRAINT          fk_patient_nurse_cares FOREIGN KEY (Nurse_ID)
                            REFERENCES NURSE(Nurse_ID),
    CONSTRAINT          fk_patient_staff_admits FOREIGN KEY(Staff_ID)
                            REFERENCES STAFF(Staff_ID)
);

CREATE TABLE NORMAL_SYMPTOM (
    Patient_no          CHAR(6),
    Normal_symptoms     VARCHAR(50),
    PRIMARY KEY         (Patient_no, Normal_symptoms)
    CONSTRAINT          fk_normal_symptom_patient_no FOREIGN KEY
    (Patient_no)
                            REFERENCES PATIENT(Patient_no)
                            ON DELETE CASCADE
);

CREATE TABLE SERIOUS_SYMPTOM (
    Patient_no          CHAR(6),
    Serious_symptoms    VARCHAR(50),
    PRIMARY KEY         (Patient_no, Serious_symptoms)
```

```
33      CONSTRAINT          fk_serious_symptom_patient_no FOREIGN KEY
        (Patient_no)
34                                  REFERENCES PATIENT(Patient_no)
35                                  ON DELETE CASCADE
36  );
37
38  CREATE TABLE PATIENT_COMORBIDITY (
39      Patient_no          CHAR(6),
40      Comorbidity         VARCHAR(50),
41      PRIMARY KEY         (Comorbidity, Patient_no),
42      CONSTRAINT          fk_comorbidity_patient_no FOREIGN KEY (Patient_no)
43                                  REFERENCES PATIENT(Patient_no)
44                                  ON DELETE CASCADE
45  );
```

- TABLE PATIENT
  - **Patient_no**: is a fixed-length string with 6 characters and set to PRIMARY KEY. For example: PAT001, PAT001, etc.
  - **Patient_name**: is a variable-length string with maximum 30 characters and is a NOT NULL attribute. Since the patient name helps the camp staffs and the patient communicate with each other easily.
  - **ID_no**: is a fix-length string with 9 characters attribute, storing their national identity card number. This attribute is NOT NULL attribute since the camp want to know more information (if neccessary) related to ID card number.
  - **Phone_no**: is a fixed-length string with 10 characters, storing the patient's phone number.
  - **Gender**: is indicated by 1-character string.
  - **Patient_address**: is a variable-length string with maximum 50 characters, indicating where the patient are living.
  - **Discharge_date**: is a DATE type attribute, indicating the discharge date from the camp of the patient.
  - **Nurse_ID**: Explain above.
  - **Admission_date**: is a DATE type attribute, indicating the admission date of the patient.
  - **Location_before**: is a variable-length string with maximum 50 characters, indicating where the patient are moved to the camp.
  - **Staff_ID**: Explain above.
  - CONSTRAINT **Check_Patient_Gender**: the same as Check_Gender.
  - CONSTRAINT **fk_patient_nurse_cares** is FOREIGN KEY referencing to Nurse_ID of NURSE table for the relationship *"Each inpatient is taken care of by a nurse; a nurse can take care of many inpatients at the same time"*.
  - CONSTRAINT **fk_patient_staff_admits** is FOREIGN KEY referencing to Staff_ID of STAFF table for the relationship *"A staff may admit many patients, and a patient is admitted by a staff"*.

- TABLES NORMAL_SYMPTOM, SERIOUS_SYMPTOM and PATIENT COMORBIDITY
  - **Patient_no**: Explain above.
  - **Normal_symptoms**, **Serious_symptoms** and **Serious_symptoms**:are variable-length strings with maximum 50 characters and are respectively combined with the **Patient_no** to form the PRIMARY KEY of NORMAL_SYMPTOM, SERIOUS_SYMPTOM and PATIENT_COMORBIDITY tables.
  - **CONSTRAINT fk_normal_symptom_patient_no**, **fk_serious_symptom_patient_no** and **fk_comorbidity_patient_no**: are FOREIGN KEY referencing to Patient_no of PATIENT table.
    ON DELETE CASCADE constraint is used to delete the rows from the NORMAL_SYMPTOM SERIOUS_SYMPTOM and PATIENT_COMORBIDITY tables automatically, when the corresponding rows from the PATIENT table are deleted.

```
CREATE TABLE ROOM (
    Room_no             CHAR(3)         PRIMARY KEY,
    Building_no         CHAR(3),
    Floor_no            SMALLINT,
    Capacity            SMALLINT,
    Room_Type           VARCHAR(12)
);
```

- TABLE ROOM
  - **Room_no**: is a fixed-length string with 3 characters and set to PRIMARY KEY. For example: 102, 304,...
  - **Building_no**: is a fixed-length string with 3 characters. For example: A04, B12,...
  - **Floor_no**: is a integer number, indicating the floor number
  - **Capacity**: is a integer numbe, indicating the maximum patient slots the room can house.
  - **Room_type**: is a variable-length string with a maximum of 12 characters, indicating the type of room.

```
CREATE TABLE ASSIGNED_TO (
    Patient_no          CHAR(6),
    Room_no             CHAR(3)         NOT NULL,
    Staff_ID            CHAR(6),
    Patient_condition   VARCHAR(30),
    PRIMARY KEY         (Patient_no, Staff_ID),
    CONSTRAINT          fk_assign_room_no FOREIGN KEY(Room_no)
                            REFERENCES ROOM(Room_no),
    CONSTRAINT          fk_assign_patient_no FOREIGN KEY(Patient_no)
                            REFERENCES PATIENT(Patient_no),
    CONSTRAINT          fk_assign_staff_ID FOREIGN KEY(Staff_ID)
                            REFERENCES STAFF(Staff_ID)
```

```
);
```

- TABLE ASSIGNED_TO
  - **Patient_no** and **Staff_ID**: are fixed-length strings with 6 characters and set to combination PRIMARY KEY.
  - **Room_no**: is a fixed-length string with 3 characters and is a NOT NULL attribute, since the camp want to know more information related to which patients are assigned to which rooms.
  - **Patient_condition**: is a variable-length string with maximum 30 characters, indicating the current condition of the patient.
  - CONSTRAINTS **fk_assign_room_no**, **fk_assign_patient_no** and **fk_assign_staff_ID** are FOREIGN KEY referencing to Room_no, Patient_no and Staff_ID of ROOM, PATIENT and STAFF tables respectively for the relationship *"When admitted by a staff, a patient is assigned into a room based on his or her current condition"*

```
CREATE TABLE PA_LOCATION (
    Patient_no          CHAR(6),
    Staff_ID            CHAR(6),
    Locations           VARCHAR(50),
    PRIMARY KEY         (Patient_no, Staff_ID, Locations),
    CONSTRAINT          fk_location_assign FOREIGN KEY (Patient_no,
    Staff_ID)
                            REFERENCES ASSIGNED_TO(Patient_no, Staff_ID)
                            ON DELETE CASCADE
);
```

- TABLE PA_LOCATION
  - **Patient_no** and **Staff_ID**: are fixed-length strings with 6 characters and set to combination PRIMARY KEY.
  - **Locations**: is a variable-length string with maximum 50 characters and also set to combination PRIMARY KEY.
  - CONSTRAINTS **fk_location_assign** is FOREIGN KEY referencing to ASSIGN_TO table. **ON DELETE CASCADE** constraint is used to delete the rows from the PA_LOCATIONS table automatically, when the corresponding rows from the ASSIGNED_TO table.

```
CREATE TABLE TREATS (
    Patient_no          CHAR(6),
    Doctor_ID           CHAR(6),
    Medication_code     CHAR(7),
    Start_date          DATE,
    End_date            DATE,
    Treat_result        VARCHAR(255),
    PRIMARY KEY         (Patient_no, Doctor_ID, Medication_code),
    CONSTRAINT          fk_treat_medication_code FOREIGN KEY
    (Medication_code)
```

```
                                REFERENCES MEDICATION(Unique_code),
    CONSTRAINT          fk_treat_doctor_id FOREIGN KEY (Doctor_ID)
                                REFERENCES DOCTOR(Doctor_ID),
    CONSTRAINT          fk_treat_patient_no FOREIGN KEY (Patient_no)
                                REFERENCES PATIENT(Patient_no)
                                ON DELETE CASCADE
);
```

- TABLE TREATS
  - **Patient_no**, **Doctor_ID** and **Medication_code**: are a fixed-length strings, which are set to combination PRIMARY KEY.
  - **Start_date** and **End_date**: are DATE type attributes, indicating the start date and end date of the treatment period respectively.
  - **Treat_result**: is a variable-length string with a maximum 255 characters, indicating the treatment result provided by the doctor.
  - **CONSTRAINT fk_treat_doctor_id**: is FOREIGN KEY referencing to Doctor_ID of DOCTOR table for the relationship *"A patient can receive treatment from at least one doctor. A doctor can treat many patients at the same time, or sometimes, he has no patients to treat".*
  - **CONSTRAINT fk_treat_medication_code**: is FOREIGN KEY referencing to Unique_code of MEDICATION table for the relationship *"The camp needs the details of each treatment such as treatment period (start date and end date), result, and medications."*
  - **CONSTRAINT fk_treat_patient_no** is FOREIGN KEY referencing to Patient_no of PATIENT table for the relationship *"The camp needs the details of each treatment such as treatment period (start date and end date), result, and medications."*
    ON DELETE CASCADE constraint is used to delete the rows from the TREATS table automatically, when the corresponding rows from the PATIENT table.

```
CREATE TABLE PATIENT_TEST_RESULTS (
    Patient_no          CHAR(6),
    SPO2                DECIMAL(5,2),
    Respiratory_rate    INT,
    PCR_test            CHAR(1),
    PCR_ct_value        INT,
    Quick_test          CHAR(1),
    Quick_ct_value      INT,
    Warning_mark        CHAR(1),
    CONSTRAINT          check_boolean CHECK ((PCR_test = 'T' or PCR_test =
    'F' or PCR_test IS NULL) and (Quick_test = 'T' or Quick_test = 'F' or
    Quick_test IS NULL) and (Warning_mark = 'T' or Warning_mark = 'F' or
    Warning_mark IS NULL)),
    PRIMARY KEY         (Patient_no, SPO2, Respiratory_rate, PCR_test,
    Quick_test),
    CONSTRAINT          fk_test_patient_no FOREIGN KEY (Patient_no)
                                REFERENCES PATIENT(Patient_no)
                                ON DELETE CASCADE
```

```
);
```

- TABLE PATIENT_TEST_RESULTS
  - **Patient_no** Explain above.
  - **SPO2** is the number with 0.01 accuracy, indicating the percent saturation of oxygen in the blood.
  - **Respiratory_rate** is a INT type attribute, indicating the number of breaths per minute.
  - **PCR_test** and **Quick_test** is 1-character strings ('T' or 'F'), indicating the test result True or False respectively.
  - **PCR_ct_value** and **Quick_ct_value** are INT type attribute, indicating the corresponding cycle threshold value in case the test result is positive.
  - **Warning_mark** is 1-character string, ('T' or 'F'), indicating whether the patient needs a healthcare action from the doctors.
  - **CONSTRAINT check_boolean** the same as Check_Gender.
  - **CONSTRAINT fk_test_patient_no**: is the FOREIGN KEY referencing to the Patient_no of the PATIENT table.
    ON DELETE CASCADE constraint is used to delete the rows from the PATIENT_TEST_RESULTS table automatically, when the corresponding rows from the PATIENT table.

2.1.2 Insert data

```
INSERT INTO PERSON VALUES('NUR001', '24 Le Loi Str, HCM City'             , 'Nguyen Van A'      , 'F', '0945519390');
INSERT INTO PERSON VALUES('NUR002', '69 Nguyen Binh Khiem Str, HCM City'  , 'Thay Giao Ba'      , 'M', '0838963738');
INSERT INTO PERSON VALUES('NUR003', '44/32 Phan Xich Long Str, HCM City'  , 'Takahashi N'       , 'M', '0904445454');
INSERT INTO PERSON VALUES('NUR004', '497 Hoa Hao Str, HCM City'           , 'Nguyen Van A'      , 'F', '0788757051');
INSERT INTO PERSON VALUES('NUR005', '168 Ly Thuong Kiet Str, HCM City'    , 'Nobi Nobita'       , 'M', '0789788124');
INSERT INTO PERSON VALUES('NUR006', '26 Le Lai Str, HCM City'             , 'Nguyen Van B'      , 'F', '0857871872');
INSERT INTO PERSON VALUES('NUR007', '6 Nguyen Trai Str, Dong Nai Province', 'Thay Giao Bon'     , 'M', '0386272293');
INSERT INTO PERSON VALUES('NUR008', '44 Nguyen Thuong Hien, HCM City'     , 'Satoshi'           , 'M', '0829397332');
INSERT INTO PERSON VALUES('NUR009', '497 30 Thang 4, Vinh Long Province'  , 'Shin Nohara'       , 'F', '0819739173');
INSERT INTO PERSON VALUES('NUR010', '168 Ly Thuong Kiet Str, Ha Noi'      , 'Nobi Doraemon'     , 'M', '0819379733');

INSERT INTO PERSON VALUES('DOC001', '1 Tran Hung Dao Str, HCM City'       , 'Tran Thi B'        , 'F', '0786543678');
```

```
13 INSERT INTO PERSON VALUES('DOC002', '268 Ly Thuong Kiet Str, HCM City'    ,
   'Nguyen Duc D'      , 'M', '0338861579');
14 INSERT INTO PERSON VALUES('DOC003', '5 Phan Xich Long Str, Da Nang City'  ,
   'Yamada N'          , 'M', '0908925454');
15 INSERT INTO PERSON VALUES('DOC004', '565 Phan Huy Ich Str, HCM City'      ,
   'Nguyen Thi Trau'   , 'F', '0972377382');
16 INSERT INTO PERSON VALUES('DOC005', '456 Le Thai To, Tien Giang Province' ,
   'Tran Van N'        , 'M', '0782396473');
17
18 INSERT INTO PERSON VALUES('STA001', '497 Hoa Hao Str, HCM City'           ,
   'Nguyen Van A'      , 'M', '0683286282');
19 INSERT INTO PERSON VALUES('STA002', '24 Ly Thai To Str, HCM City'         ,
   'Le Duan'           , 'M', '0789253282');
20 INSERT INTO PERSON VALUES('STA003', '57 Phan Xich Long Str, HCM City'     ,
   'Takeshi A'         , 'F', '0917637429');
21 INSERT INTO PERSON VALUES('STA004', '123 Nguyen Kim Str, HCM City'        ,
   'Nguyen Huu Truong', 'F', '0891638217');
22 INSERT INTO PERSON VALUES('STA005', '98 Le Thanh Tong Str, HCM City'      ,
   'Le Quang Tin'      , 'M', '0978133782');
23 INSERT INTO PERSON VALUES('STA006', '124 Nguyen Kim Str, HCM City'        ,
   'Nguyen Minh Tho'   , 'F', '0829379372');
24 INSERT INTO PERSON VALUES('STA007', '676 Le Thanh Tong Str, Ha Noi'       ,
   'Ly Quang Tin'      , 'M', '0872903782');
25
26 INSERT INTO PERSON VALUES('VOL001', '215 Ngo Gia Tu Str, HCM City'        ,
   'Nguyen Van A'      , 'M', '0965487364');
27 INSERT INTO PERSON VALUES('VOL002', '781 3 thang 2 Str, HCM City'         ,
   'Luong Van T'       , 'F', '0897685362');
28 INSERT INTO PERSON VALUES('VOL003', '4 Phan Xich Long Str, HN City'       ,
   'Honda N'           , 'F', '0897658436');
29 INSERT INTO PERSON VALUES('VOL004', '15 Ngo Gia Tu Str, HCM City'         ,
   'Nguyen Van B'      , 'M', '0812328123');
30 INSERT INTO PERSON VALUES('VOL005', '119 Ngo Quyen Str, HCM City'         ,
   'Nguyen Van Binh'   , 'M', '0987387388');
31
32 INSERT INTO PERSON VALUES('MAN001', '20 Cach Mang Thang Tam Str, HCM City',
   'Nguyen Van A'      , 'M', '0763872917');
33 INSERT INTO PERSON VALUES('MAN002', '180 Nguyen Anh Thu Str, HCM City'    ,
   'Nha Bach D'        , 'F', '0989873279');
34 INSERT INTO PERSON VALUES('MAN003', '32 Tran Hung Dao Str, HCM City'      ,
   'Nakamura R'        , 'M', '0989897678');
35 INSERT INTO PERSON VALUES('MAN004', '44 Le Loi Str, HCM City'             ,
   'Tran Van Z'        , 'F', '0943567854');
36 INSERT INTO PERSON VALUES('MAN005', '12 Bach Dang Str, HCM City'          ,
   'Nguyen Thi Ba'     , 'M', '0976767578');
37
38 INSERT INTO NURSE VALUES('NUR001');
39 INSERT INTO NURSE VALUES('NUR002');
```

```
40 INSERT INTO NURSE VALUES('NUR003');
41 INSERT INTO NURSE VALUES('NUR004');
42 INSERT INTO NURSE VALUES('NUR005');
43 INSERT INTO NURSE VALUES('NUR006');
44 INSERT INTO NURSE VALUES('NUR007');
45 INSERT INTO NURSE VALUES('NUR008');
46 INSERT INTO NURSE VALUES('NUR009');
47 INSERT INTO NURSE VALUES('NUR010');
48
49 INSERT INTO DOCTOR VALUES('DOC001');
50 INSERT INTO DOCTOR VALUES('DOC002');
51 INSERT INTO DOCTOR VALUES('DOC003');
52 INSERT INTO DOCTOR VALUES('DOC004');
53 INSERT INTO DOCTOR VALUES('DOC005');
54
55 INSERT INTO STAFF VALUES('STA001');
56 INSERT INTO STAFF VALUES('STA002');
57 INSERT INTO STAFF VALUES('STA003');
58 INSERT INTO STAFF VALUES('STA004');
59 INSERT INTO STAFF VALUES('STA005');
60 INSERT INTO STAFF VALUES('STA006');
61 INSERT INTO STAFF VALUES('STA007');
62
63 INSERT INTO VOLUNTEER VALUES('VOL001');
64 INSERT INTO VOLUNTEER VALUES('VOL002');
65 INSERT INTO VOLUNTEER VALUES('VOL003');
66 INSERT INTO VOLUNTEER VALUES('VOL004');
67 INSERT INTO VOLUNTEER VALUES('VOL005');
68
69 INSERT INTO MANAGER VALUES('MAN001','DOC001');
70 INSERT INTO MANAGER(Manager_ID) VALUES('MAN002');
71 INSERT INTO MANAGER(Manager_ID) VALUES('MAN003');
72 INSERT INTO MANAGER(Manager_ID) VALUES('MAN004');
73 INSERT INTO MANAGER(Manager_ID) VALUES('MAN005');
74
75 INSERT INTO MEDICATION VALUES('MED0001', 'Bromhexin hydroclorid', 'cough'
   , 2000, '20-10-2022');
76 INSERT INTO MEDICATION VALUES('MED0002', 'Paracetamol'          , 'fever,
   headache', 3000, '08-11-2022');
77 INSERT INTO MEDICATION VALUES('MED0003', 'Terpin Codein'        , 'cough'
   , 1000, '02-10-2022');
78 INSERT INTO MEDICATION VALUES('MED0004', 'Esphelaran'           ,
   'constipation'   , 3000, '20-10-2022');
79 INSERT INTO MEDICATION VALUES('MED0005', 'Mythyl Orange'        , 'cold'
   , 2500, '22-12-2022');
80
```

```sql
INSERT INTO PATIENT VALUES('PAT001', 'Peter Tran'        , '323692071',
'0948297879', 'M', 'Bien Hoa City, Dong Nai Province'  , '2020-11-20',
'NUR001', '20-10-2020', 'Bach Khoa Dormitory'          , 'STA001');
INSERT INTO PATIENT VALUES('PAT002', 'Ben Parker'        , '573723683',
'0893797192', 'F', '268 Ly Thuong Kiet Str, HCM City'  , '2020-10-20',
'NUR002', '25-09-2020', 'Bach Khoa University'         , 'STA002');
INSERT INTO PATIENT VALUES('PAT003', 'Nguyen Bach Cuc'  , '871238978',
'0997187933', 'F', '67 Tran Hung Dao Str, HCM City'    ,  NULL       ,
'NUR003', '01-11-2021', '23 Tran Hung Dao Str, HCM City', 'STA003');
INSERT INTO PATIENT VALUES('PAT004', 'Le The Duy'        , '882772739',
'0917936782', 'M', '127 Tran Quoc Toan Str, HCM City'  , '2021-04-03',
'NUR005', '25-03-2021', 'University of Economics'      , 'STA005');
INSERT INTO PATIENT VALUES('PAT005', 'Truong Quang Thai', '183934678',
'0891731933', 'F', '635 Ly Thuong Kiet Str, HCM City'  , '2021-06-30',
'NUR004', '23-06-2021', 'Ly Thuong Kiet Co.op Mart'    , 'STA006');
INSERT INTO PATIENT VALUES('PAT006', 'Nguyen Van A'     , '723972033',
'0781673811', 'F', '36 Nguyen Tri Phuong Str, HCM City',  NULL       ,
'NUR006', '25-10-2021', 'Tu Du Hospital'               , 'STA007');
INSERT INTO PATIENT VALUES('PAT007', 'Tran Van B'        , '913797193',
'0367481791', 'M', '23 Hai Ba Truong Str, HCM City'    , '2021-05-03',
'NUR008', '30-03-2021', 'Tien Giang Province'          , 'STA004');
INSERT INTO PATIENT VALUES('PAT008', 'Nguyen Van A'     , '742810139',
'0183981613', 'M', '33 Dinh Tien Hoang Str, HCM City'  , '2021-09-02',
'NUR009', '07-08-2021', 'United State'                 , 'STA002');
INSERT INTO PATIENT VALUES('PAT009', 'Le Thi C'          , '816387793',
'0819738618', 'F', '29 Phu Dong Str, HCM City'         ,  NULL       ,
'NUR007', '15-10-2021', 'Bach Khoa Dormitory'          , 'STA003');
INSERT INTO PATIENT VALUES('PAT010', 'Bui Trong D'       , '183834678',
'0891379613', 'M', '99 Ngo Quyen Str, HCM City'        , '2021-05-07',
'NUR001', '01-05-2021', '56 Nguyen Trai Str, HCM City' , 'STA001');
INSERT INTO PATIENT VALUES('PAT011', 'Ly Van T'          , '813781731',
'0185714617', 'F', '94 Au Co Str, HCM City'            , '2021-05-04',
'NUR002', '20-04-2021', '6 Ly Tu Trong Str, HCM City'  , 'STA002');
INSERT INTO PATIENT VALUES('PAT012', 'Tran Van V'        , '123645631',
'0312412879', 'M', '8 Tan Thuan Dong Str, HCM City'    , '2021-05-20',
'NUR001', '25-12-2021', 'Bach Khoa Dormitory'          , 'STA001');


INSERT INTO NORMAL_SYMPTOM VALUES('PAT001', 'dry cough');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT001', 'tiredness');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT001', 'loss of smell');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT002', 'fever');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT002', 'cough');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT003', 'discolouration of fingers');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT003', 'sore throat');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT004', 'aches and pains');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT004', 'diarrhoea');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT004', 'conjunctivitis');
```

```
INSERT INTO NORMAL_SYMPTOM VALUES('PAT005', 'loss of taste');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT005', 'cough');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT006', 'discolouration of toes');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT007', 'sore throat');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT007', 'fever');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT008', 'tiredness');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT008', 'loss of smell');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT008', 'fever');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT009', 'cough');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT009', 'discolouration of fingers');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT009', 'headache');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT010', 'rash on skin');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT010', 'discolouration of toes');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT010', 'fever');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT011', 'sore throat');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT011', 'dry cough');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT011', 'tiredness');
INSERT INTO NORMAL_SYMPTOM VALUES('PAT011', 'headache');

INSERT INTO SERIOUS_SYMPTOM VALUES('PAT001', 'shortness of breath');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT001', 'chest pain');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT002', 'loss of speech');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT002', 'shortness of breath');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT004', 'chest pain');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT005', 'shortness of breath');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT006', 'chest pain');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT008', 'loss of movement');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT009', 'chest pain');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT010', 'loss of movement');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT011', 'shortness of breath');
INSERT INTO SERIOUS_SYMPTOM VALUES('PAT011', 'chest pain');

INSERT INTO PATIENT_COMORBIDITY VALUES('PAT001', 'cancer');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT002', 'chronic lung diseases');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT003', 'diabetes');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT005', 'chronic lung diseases');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT005', 'cancer');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT006', 'immunocompromised
state');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT007', 'heart conditions');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT008', 'immunocompromised
state');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT010', 'heart conditions');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT010', 'diabetes');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT011', 'diabetes');
INSERT INTO PATIENT_COMORBIDITY VALUES('PAT011', 'cancer');

INSERT INTO ROOM VALUES ('200', 'B12', 2, 10, 'VIP');
```

```sql
INSERT INTO ROOM VALUES ('101', 'B12', 1, 30, 'Normal');
INSERT INTO ROOM VALUES ('102', 'B12', 1, 50, 'Normal');
INSERT INTO ROOM VALUES ('103', 'B12', 1, 40, 'Normal');
INSERT INTO ROOM VALUES ('304', 'D07', 3, 50, 'Normal');
INSERT INTO ROOM VALUES ('405', 'A04', 4, 20, 'Recuperation');
INSERT INTO ROOM VALUES ('406', 'A04', 4, 20, 'Recuperation');
INSERT INTO ROOM VALUES ('407', 'A04', 4, 20, 'Recuperation');
INSERT INTO ROOM VALUES ('308', 'C10', 3, 10, 'VIP');
INSERT INTO ROOM VALUES ('509', 'D08', 5, 50, 'Normal');

INSERT INTO ASSIGNED_TO VALUES ('PAT001', '101', 'STA001', 'Normal');
INSERT INTO ASSIGNED_TO VALUES ('PAT002', '101', 'STA002', 'Normal');
INSERT INTO ASSIGNED_TO VALUES ('PAT003', '200', 'STA003', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT004', '308', 'STA003', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT005', '102', 'STA003', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT006', '405', 'STA005', 'Normal');
INSERT INTO ASSIGNED_TO VALUES ('PAT007', '407', 'STA006', 'Normal');
INSERT INTO ASSIGNED_TO VALUES ('PAT008', '103', 'STA007', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT009', '102', 'STA004', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT010', '308', 'STA002', 'Serious');
INSERT INTO ASSIGNED_TO VALUES ('PAT011', '509', 'STA005', 'Normal');

INSERT INTO PA_LOCATION VALUES('PAT001', 'STA001', 'Emergency room');
INSERT INTO PA_LOCATION VALUES('PAT002', 'STA002', 'Normal room');
INSERT INTO PA_LOCATION VALUES('PAT003', 'STA003', 'Recuperation room');
INSERT INTO PA_LOCATION VALUES('PAT004', 'STA003', 'Normal room');
INSERT INTO PA_LOCATION VALUES('PAT005', 'STA003', 'Emergency room');
INSERT INTO PA_LOCATION VALUES('PAT006', 'STA005', 'Normal room');
INSERT INTO PA_LOCATION VALUES('PAT007', 'STA006', 'Recuperation room');
INSERT INTO PA_LOCATION VALUES('PAT008', 'STA007', 'Normal room');
INSERT INTO PA_LOCATION VALUES('PAT009', 'STA004', 'Normal room');
INSERT INTO PA_LOCATION VALUES('PAT010', 'STA002', 'Recuperation room');
INSERT INTO PA_LOCATION VALUES('PAT011', 'STA005', 'Normal room');

INSERT INTO TREATS VALUES ('PAT001', 'DOC001', 'MED0001', '2022-10-20',
'27-10-2022', 'Normal');
INSERT INTO TREATS VALUES ('PAT001', 'DOC001', 'MED0002', '2020-10-28',
'05-10-2020', 'Normal');
INSERT INTO TREATS VALUES ('PAT001', 'DOC001', 'MED0003', '2021-10-06',
'20-10-2021', 'Normal');
INSERT INTO TREATS VALUES ('PAT002', 'DOC002', 'MED0004', '2020-09-25',
'02-10-2020', 'Normal');
INSERT INTO TREATS VALUES ('PAT002', 'DOC002', 'MED0005', '2021-10-03',
'10-10-2021', 'Normal');
INSERT INTO TREATS VALUES ('PAT002', 'DOC002', 'MED0001', '2021-10-11',
'01-11-2021', 'Normal');
INSERT INTO TREATS VALUES ('PAT003', 'DOC002', 'MED0002', '2021-11-01',
'11-11-2021', 'Normal');
```

```
192 INSERT INTO TREATS VALUES ('PAT004', 'DOC003', 'MED0005', '2021-03-25',
    '31-03-2021', 'Serious');
193 INSERT INTO TREATS VALUES ('PAT004', 'DOC001', 'MED0001', '2021-04-01',
    '03-04-2021', 'Normal');
194 INSERT INTO TREATS VALUES ('PAT005', 'DOC002', 'MED0002', '2021-06-23',
    '25-06-2021', 'Serious');
195 INSERT INTO TREATS VALUES ('PAT005', 'DOC002', 'MED0003', '2021-06-26',
    '28-06-2021', 'Serious');
196 INSERT INTO TREATS VALUES ('PAT005', 'DOC002', 'MED0004', '2021-06-29',
    '30-06-2021', 'Normal');
197 INSERT INTO TREATS VALUES ('PAT006', 'DOC003', 'MED0005', '2021-10-25',
    '25-11-2021', 'Normal');
198 INSERT INTO TREATS VALUES ('PAT006', 'DOC003', 'MED0001', '2021-10-20',
    '27-10-2021', 'Normal');
199 INSERT INTO TREATS VALUES ('PAT007', 'DOC004', 'MED0002', '2021-03-30',
    '15-04-2021', 'Normal');
200 INSERT INTO TREATS VALUES ('PAT007', 'DOC004', 'MED0003', '2021-04-16',
    '03-05-2021', 'Normal');
201 INSERT INTO TREATS VALUES ('PAT008', 'DOC005', 'MED0004', '2021-08-07',
    '27-08-2021', 'Normal');
202 INSERT INTO TREATS VALUES ('PAT008', 'DOC005', 'MED0005', '2021-09-28',
    '02-09-2021', 'Normal');
203 INSERT INTO TREATS VALUES ('PAT009', 'DOC005', 'MED0001', '2021-10-15',
    '20-10-2021', 'Serious');
204 INSERT INTO TREATS VALUES ('PAT009', 'DOC005', 'MED0002', '2021-10-21',
    '31-10-2021', 'Serious');
205 INSERT INTO TREATS VALUES ('PAT010', 'DOC003', 'MED0003', '2021-05-01',
    '02-05-2021', 'Serious');
206 INSERT INTO TREATS VALUES ('PAT010', 'DOC002', 'MED0004', '2021-05-03',
    '04-05-2021', 'Normal');
207 INSERT INTO TREATS VALUES ('PAT010', 'DOC003', 'MED0005', '2021-05-05',
    '07-05-2021', 'Normal');
208 INSERT INTO TREATS VALUES ('PAT011', 'DOC001', 'MED0001', '2021-04-20',
    '27-04-2021', 'Normal');
209 INSERT INTO TREATS VALUES ('PAT011', 'DOC002', 'MED0002', '2021-04-28',
    '30-04-2021', 'Normal');
210 INSERT INTO TREATS VALUES ('PAT011', 'DOC003', 'MED0003', '2021-05-01',
    '04-05-2021', 'Normal');
211
212 INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT001', 97.88, 20, 'F', NULL,
    'F', NULL, 'F');
213 INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT002', 99.12, 23, 'T', 35  ,
    'T', 36  , 'F');
214 INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT003', 50.14, 25, 'F', NULL,
    'F', NULL, 'T');
215 INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT004', 97.88, 19, 'T', 40  ,
    'T', 50  , 'F');
```

```
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT005', 86.34, 17, 'T', 32  , 'T', 40  , 'F');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT006', 50.88, 23, 'F', NULL, 'F', NULL, 'T');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT007', 98.34, 18, 'F', NULL, 'F', NULL, 'F');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT008', 97.54, 19, 'F', NULL, 'F', NULL, 'F');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT009', 50.78, 21, 'F', NULL, 'F', NULL, 'T');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT010', 96.34, 21, 'T', 33  , 'T', 34  , 'F');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT011', 89.78, 22, 'F', NULL, 'F', NULL, 'F');
INSERT INTO PATIENT_TEST_RESULTS VALUES ('PAT012', 90.69, 22, 'T', 30, 'T', 30,   'T');
```

2.2 Store Procedure / Function / SQL

2.2.1 UPDATE query

Requirement: Update patient PCR test to positive with null cycle threshold value for all patients whose admission date is from 01/09/2020.
We set the `PCR_test` attribute to `'T'` (meaning positive or true) and the `PCR_ct_value` attribute to `NULL`.

```
UPDATE PATIENT_TEST_RESULTS
SET PCR_test = 'T',
    PCR_ct_value =  NULL
WHERE EXISTS (SELECT Admission_date FROM PATIENT WHERE Admission_date >= '1-9-2020')
;
```

```
11 rows updated.
```

Based on our database, there are 11 rows updated. Check again with the command:

```
SELECT *
FROM PATIENT_TEST_RESULTS
;
```

PATIENT_NO	SPO2	RESPIRATORY_RATE	PCR_TEST	PCR_CT_VALUE	QUICK_TEST	QUICK_CT_VALUE	WARNING_MARK
PAT001	97.88	20	T	(null)	F	(null)	F
PAT002	99.12	23	T	(null)	T	36	F
PAT003	50.14	25	T	(null)	F	(null)	T
PAT004	97.88	19	T	(null)	T	50	F
PAT005	86.34	17	T	(null)	T	40	F
PAT006	50.88	23	T	(null)	F	(null)	T
PAT007	98.34	18	T	(null)	F	(null)	F
PAT008	97.54	19	T	(null)	F	(null)	F
PAT009	50.78	21	T	(null)	F	(null)	T
PAT010	96.34	21	T	(null)	T	34	F
PAT011	89.78	22	T	(null)	F	(null)	F
PAT012	90.69	22	T	30	T	30	T

Only patient `PAT012` whose `Admission_date` is `20-05-2020` has not been updated.

2.2.2 SELECT query

Requirement: Select all the patient information whose name is 'Nguyen Van A'.

```
SELECT *
FROM PATIENT
WHERE Patient_name = 'Nguyen Van A'
;
```

Note: The figure below is a .pdf file, so if the text is too small or blurry, please zoom in for a better view.

PATIENT_NO	PATIENT_NAME	ID_NO	PHONE_NO	GENDER	PATIENT_ADDRESS	DISCHARGE_DATE	NURSE_ID	ADMISSION_DATE	LOCATION_BEFORE	STAFF_ID
PAT006	Nguyen Van A	723972033	0781673811	F	36 Nguyen Tri Phuong Str, HCM City	(null)	NUR006	25-10-2021	Tu Du Hospital	STA007
PAT008	Nguyen Van A	742810139	0183981613	M	33 Dinh Tien Hoang Str, HCM City	02-09-2021	NUR009	07-08-2021	United State	STA002

2.2.3 Function

Requirement: Write a function to calculate the testing for each patient.

- Input: Patient ID.
- Output: A list of testing.

```
CREATE OR REPLACE FUNCTION get_detail_patient
(Patient_ID in PATIENT_TEST_RESULTS.Patient_no%TYPE)
 RETURN SYS_REFCURSOR
AS
   o_cursor   SYS_REFCURSOR;
BEGIN
    OPEN o_cursor FOR
        SELECT *
        FROM PATIENT_TEST_RESULTS
        WHERE PATIENT_TEST_RESULTS.Patient_no = Patient_ID;
    RETURN o_cursor;
```

```
END
;
```

```
SELECT get_detail_patient('PAT001') FROM DUAL;
```

```
{<PATIENT_NO=PAT001,SPO2=96.69,RESPIRATORY_RATE=20,PCR_TEST=F,
  PCR_CT_VALUE=null,QUICK_TEST=F,QUICK_CT_VALUE=null,WARNING_MARK
  =F>,<PATIENT_NO=PAT001,SPO2=97.88,RESPIRATORY_RATE=20,PCR_TEST=
  T,PCR_CT_VALUE=null,QUICK_TEST=F,QUICK_CT_VALUE=null,
  WARNING_MARK=F>,}
```

2.2.4 Procedure

Requirement: Write a procedure to sort the nurses in decreasing number of patients he/she takes care in a period of time.

- Input: Start date, End date.
- Output: A list of sorting nurses.

We create a PROCEDURE `sort_nurse_desc`:

```
CREATE OR REPLACE PROCEDURE sort_nurse_desc
(start_date IN DATE, end_date IN DATE, cursorParam OUT SYS_REFCURSOR)
IS
  BEGIN
    OPEN cursorParam FOR
        SELECT Nurse_ID AS Nurse, COUNT(Patient_no) AS Num_patient
        FROM PATIENT
        WHERE Admission_date >= start_date AND (Discharge_date <= end_date
        OR Discharge_date IS NULL)
        GROUP BY Nurse_ID
        ORDER BY COUNT(Patient_no) DESC;
END
;
```

Here we choose `start_date='2020-1-1'` and `end_date='2022-1-1'`. To run it, we can use these commands:

```
var c refcursor;
execute sort_nurse_desc('2020-1-1','2022-1-1',:c)
print c;
```

```
NURSE  NUM_PATIENT
------ -----------
NUR001           3
```

```
NUR002            2
NUR009            1
NUR007            1
NUR004            1
NUR006            1
NUR003            1
NUR005            1
NUR008            1
9 rows selected.
```

The output should display the total `NUM_PATIENT` equal to 12.

2.3 Building Application

The entire source code of this assignment is available here.

- In Oracle SQL Developer, our database is `QCD` (Quarantine Camp Database).
- In XAMPP (MariaDB), our database is `qcd`.

2.3.1 Create user

Requirement: Log in to the database with `DBA` privileges such as `SYS / SYSTEM` ...., create a user named `Manager` and assign all access rights to this user.

2.3.1.a Oracle SQL Developer

From `SYS` we create a `QCD_MANAGER` user and grant it `DBA` permission.

```
1  CREATE USER QCD_MANAGER IDENTIFIED BY 1;
2  GRANT CREATE SESSION TO QCD_MANAGER;
3  GRANT DBA TO QCD_MANAGER;
```

```
User QCD_MANAGER created.
Grant succeeded.
Grant succeeded.
```

From `QCD` we grant `DBA` to `QCD_MANAGER`.

```
1  GRANT DBA TO QCD_MANAGER;
```

```
Grant succeeded.
```

From now, `QCD_MANAGER` can manager `QCD` Database.

```
1  SELECT
2      D.Doctor_ID AS "Doctor ID",
3      P.Person_name AS "Doctor Name",
4      P.Phone_no AS "Phone Number",
5      P.Gender,
```

```
    P.Person_address AS "Address"
FROM QCD.DOCTOR D JOIN QCD.PERSON P ON (D.Doctor_ID = P.Person_ID)
;
```

Doctor ID	Doctor Name	Phone Number	GENDER	Address
DOC001	Tran Thi B	0786543678	F	1 Tran Hung Dao Str, HCM City
DOC002	Nguyen Duc D	0338861579	M	268 Ly Thuong Kiet Str, HCM City
DOC003	Yamada N	0908925454	M	5 Phan Xich Long Str, Da Nang City
DOC004	Nguyen Thi Trau	0972377382	F	565 Phan Huy Ich Str, HCM City
DOC005	Tran Van N	0782396473	M	456 Le Thai To, Tien Giang Province

2.3.1.b MariaDB

In phpMyAdmin, we create a table `user_manager`:

```
CREATE TABLE `user_manager` (
  `id` int(4) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `user_manager`
  ADD PRIMARY KEY (`id`);
COMMIT;
```

Then we create an account with:

- id=1
- username='admin'
- password='123456'

```
INSERT INTO `user_manager` (`id`, `username`, `password`) VALUES
(1, 'admin', '123456');
```

We use this account to login to our database in `index.php`.

```
<?php
$ret=mysqli_query($con,"SELECT * FROM user_manager WHERE
username='".$_POST['username']."' and password='".$_POST['password']."'");
  $num=mysqli_fetch_array($ret);

  if($num>0)
  {
    $extra="dashboard.php";//
    $_SESSION['login']=$_POST['username'];
    $_SESSION['id']=$num['id'];
    $host=$_SERVER['HTTP_HOST'];
    $uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
```

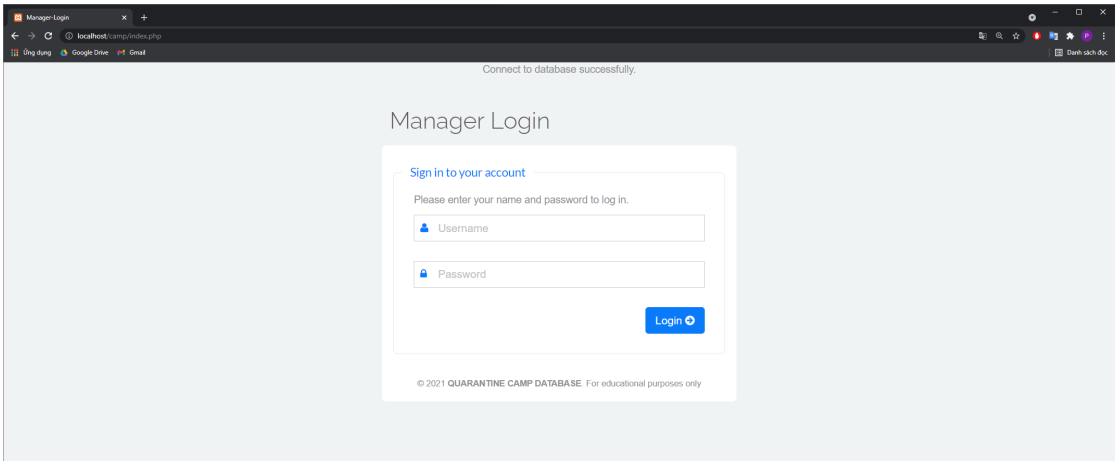
```
12      header("location:http://$host$uri/$extra");
13      exit();
14    }
15    else
16    {
17      $_SESSION['errmsg']="Invalid username or password.";
18      $extra="index.php";
19      $host  = $_SERVER['HTTP_HOST'];
20      $uri  = rtrim(dirname($_SERVER['PHP_SELF']),'/\\');
21      header("location:http://$host$uri/$extra");
22      exit();
23    }
24  ?>
```

2.3.2 Requirement function

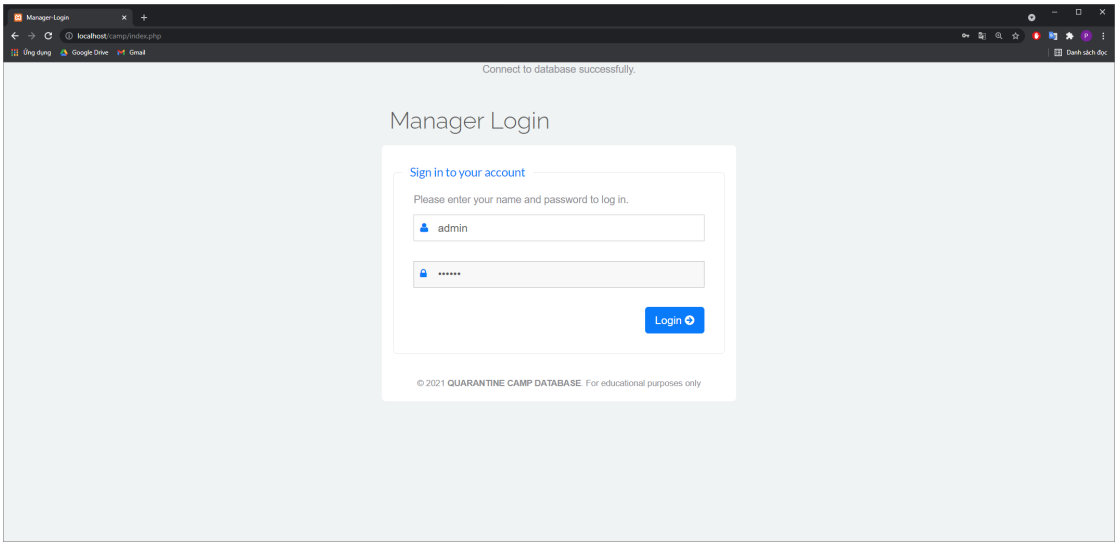
2.3.2.a Log in / Log out function

Requirement: Log in, log out (enter the user name/password for Manager account to log in/out).

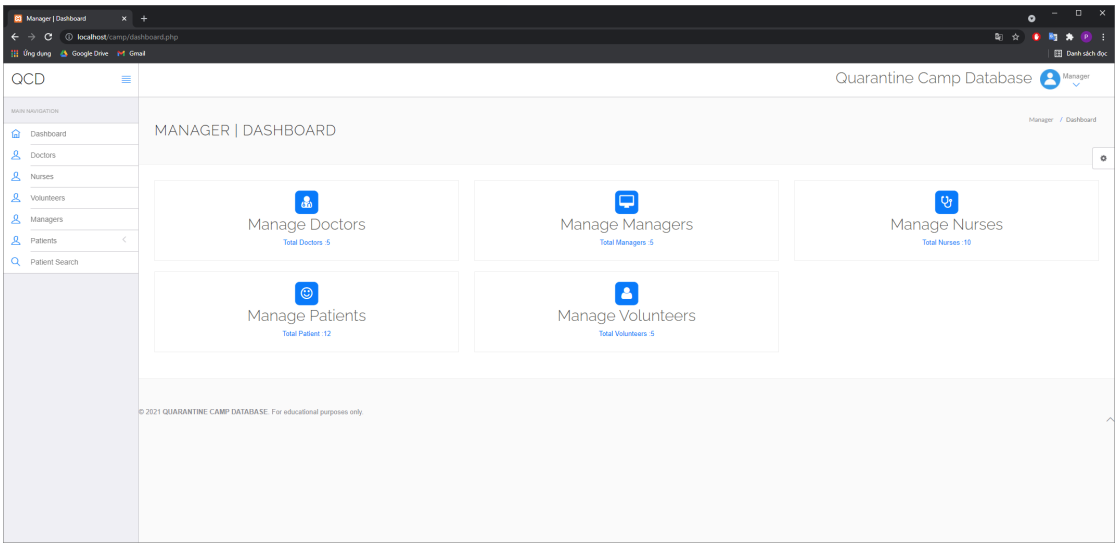
- Our interface for login function:



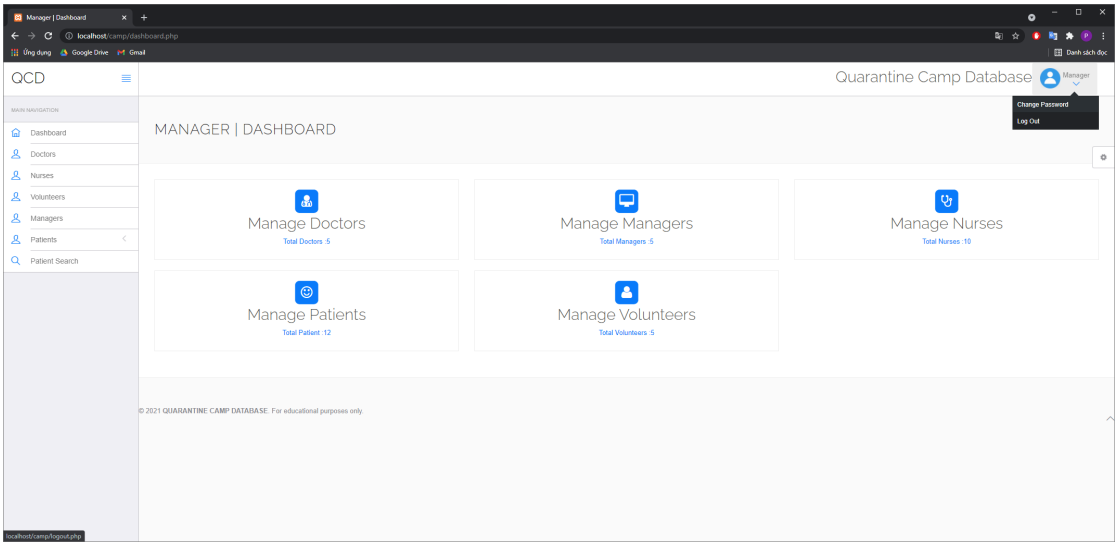
- Note that if the connection to the database is established successfully, there will be a message in the top center. And the same for the failed case. Here we login by using account `admin`:



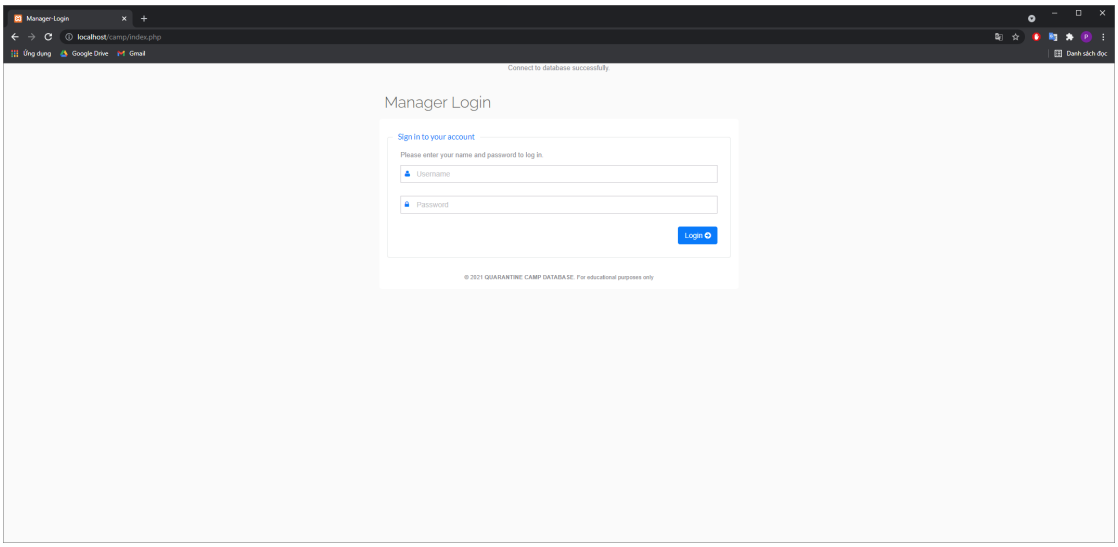
- Click on `Login` buton, if the user is successfully logged in then we can see the dashboard. Otherwise, it will display `Invalid username or password.` Luckily we can see the dashboard as well in the figure below:



- To log out, the user clicks on the Manager icon in the upper left and then selects `Log Out`.



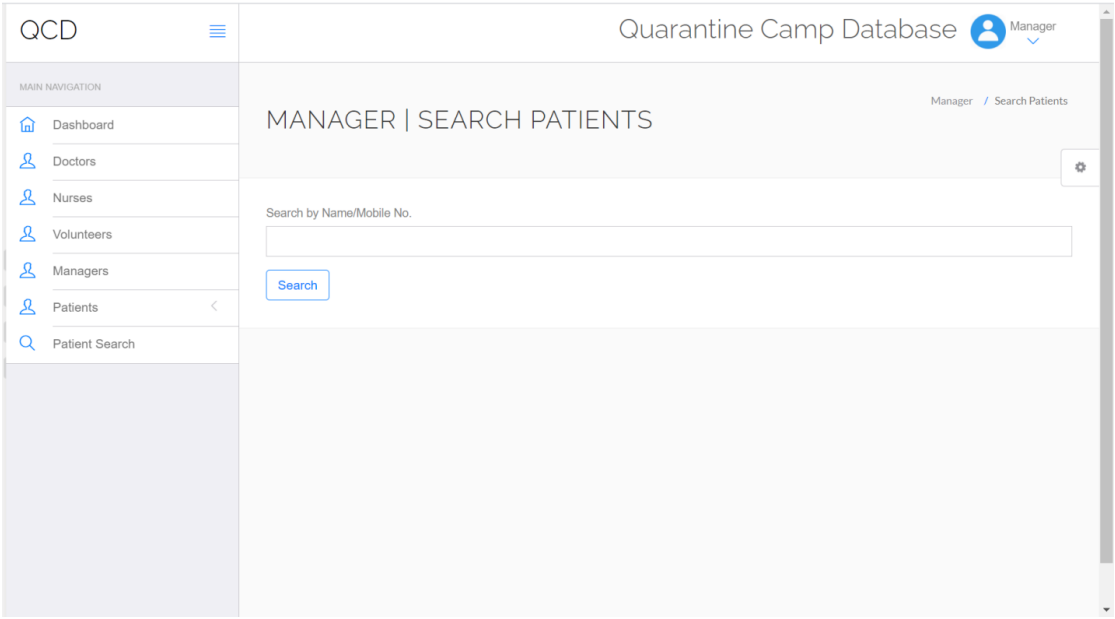
- Then we back to `index.php`:



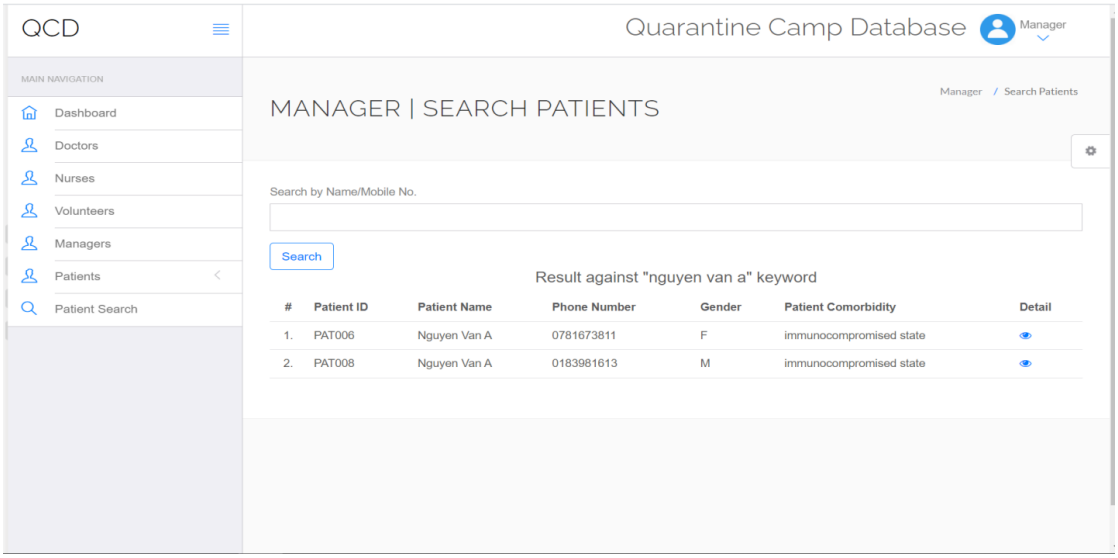
2.3.2.b Search function

Requirement: Search patient information: Search results include the name, phone number and information about his/her comorbidities.

- First we choose `Patient Search` located at the last line in the toolbar on the left, and we will get the interface as shown in the image below.

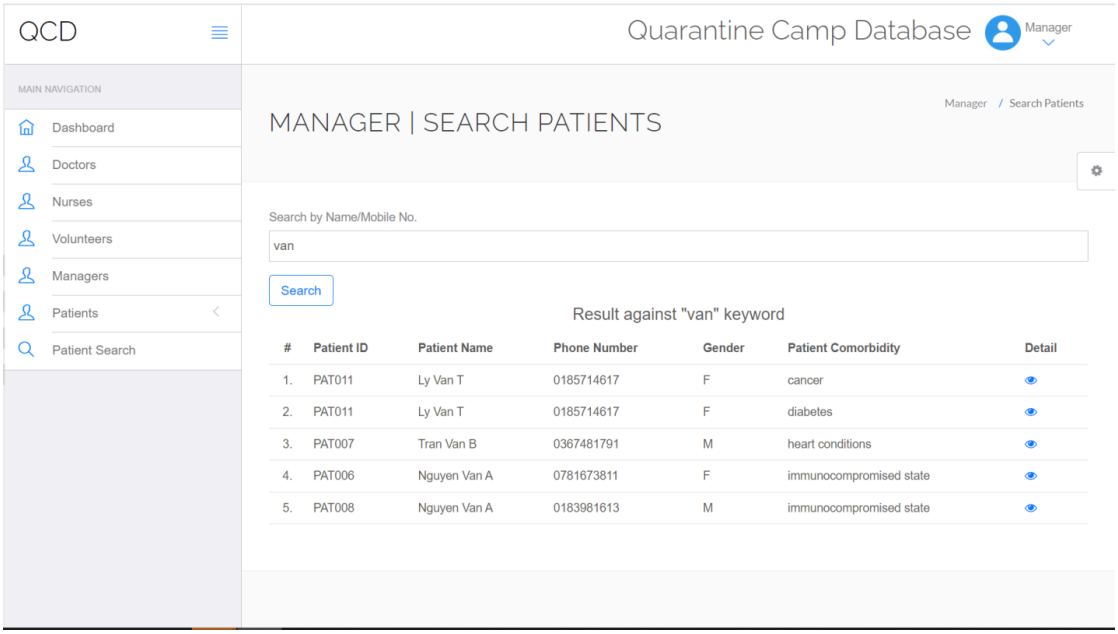


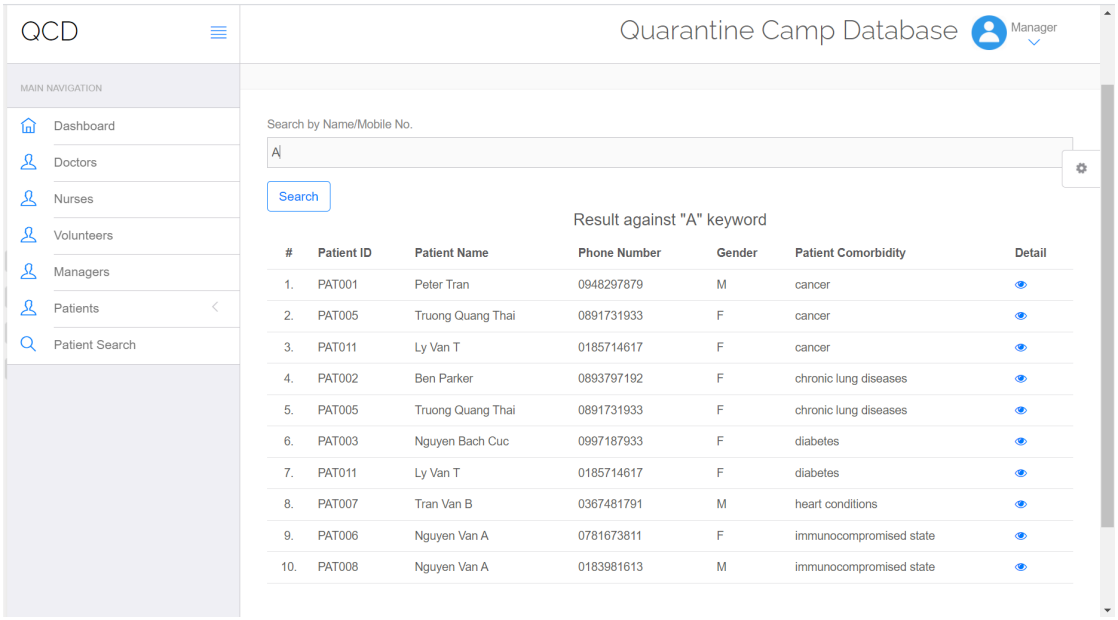
- Then select the blank line and enter information such as the patient's name or phone number to search. For example, if you want to find a patient with the full name `Nguyen Van A`, after entering the name, information about name, phone number and information about his/her comorbidities will be displayed as shown below.

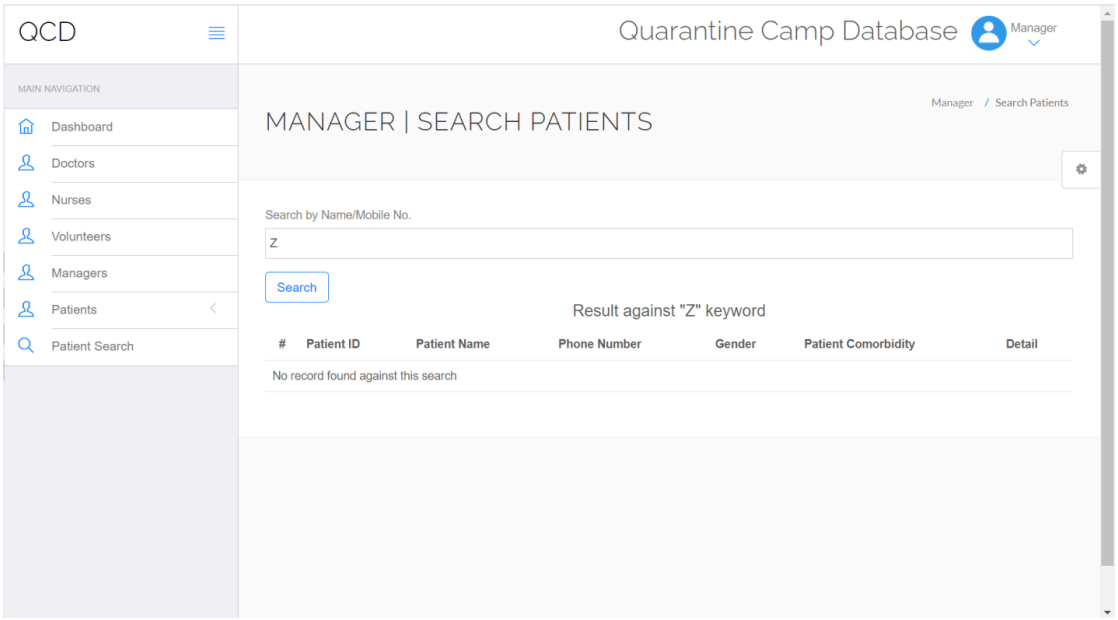


- If you can not remember the full name but only remember the last name or 1 character, that is okay, just enter it and the system will find out who has that character in their name. For example, if you only remember the patient `Nguyen Van A` with the name `Van` in the name, just search for the word `"Van"` and you will get patients with `"Van"` in the name or you just remember each of the names. If that person has the letter A, when clicking on it,

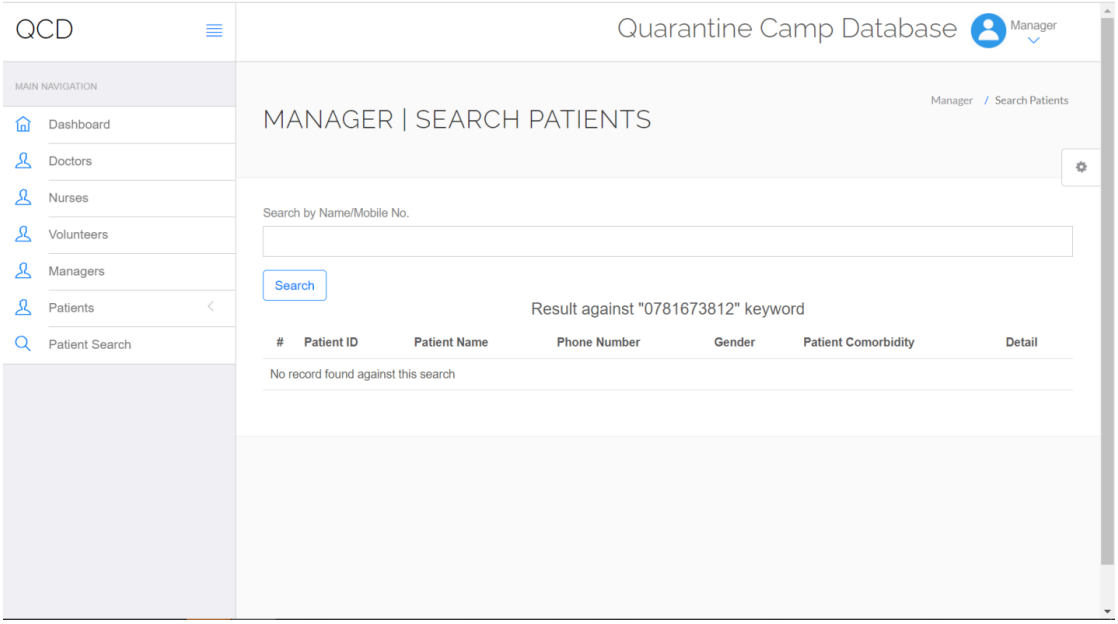
the system will display all the names with the letter A in the name. If the search fails the system will output the text “No record found against this search”.







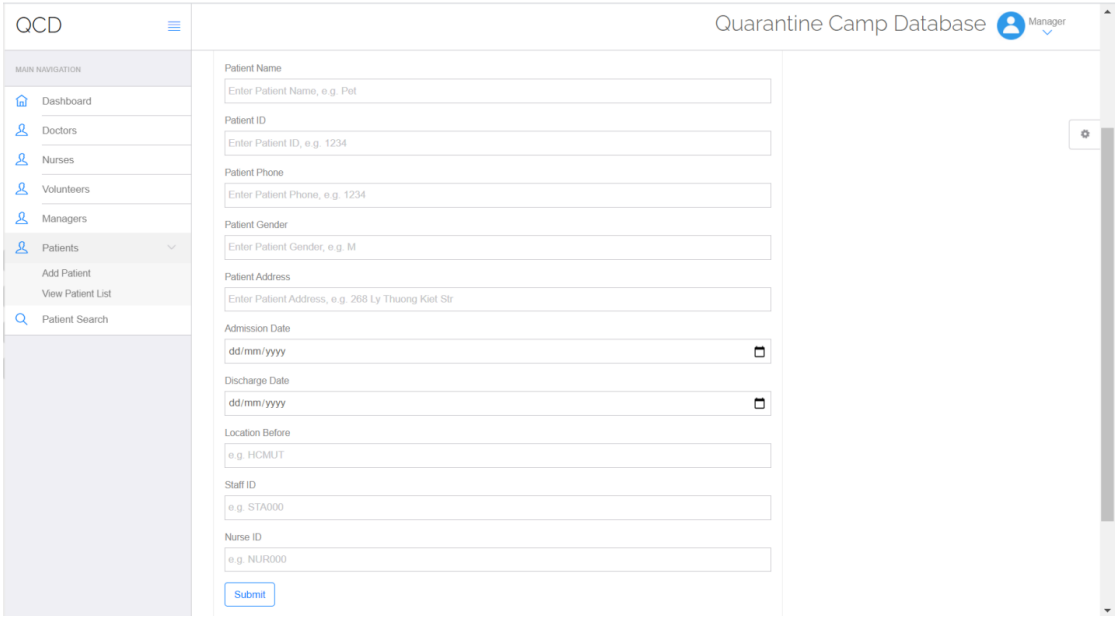
- But for phones, you must enter it correctly and completely otherwise the system will output "No record found against this search".



2.3.2.c Add function

Requirement: Add information for a new patient.

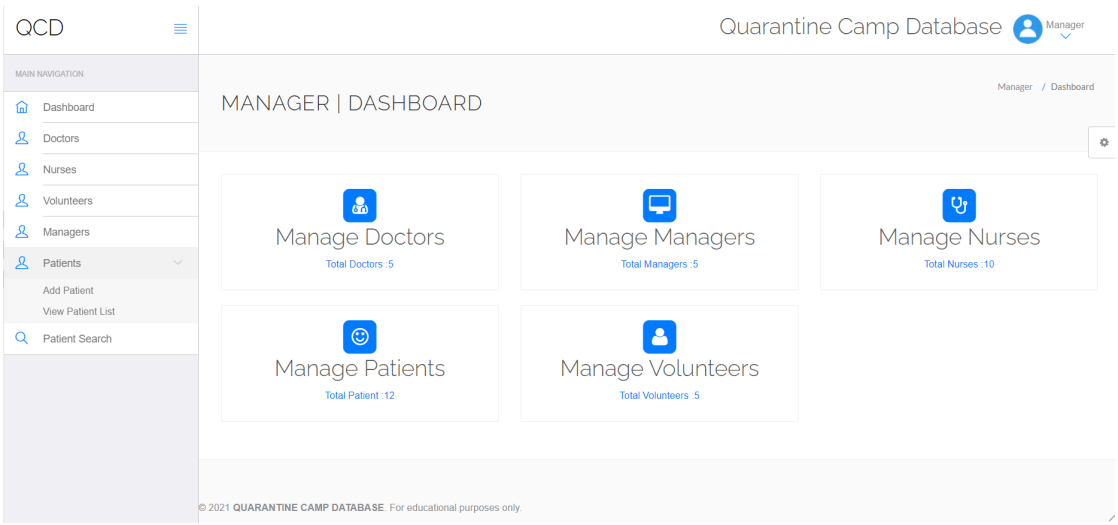
- Select `Patient` on the last line in the left toolbar, after appearing 2 command lines, select `Add Patient` in the first line. We get the following interface.

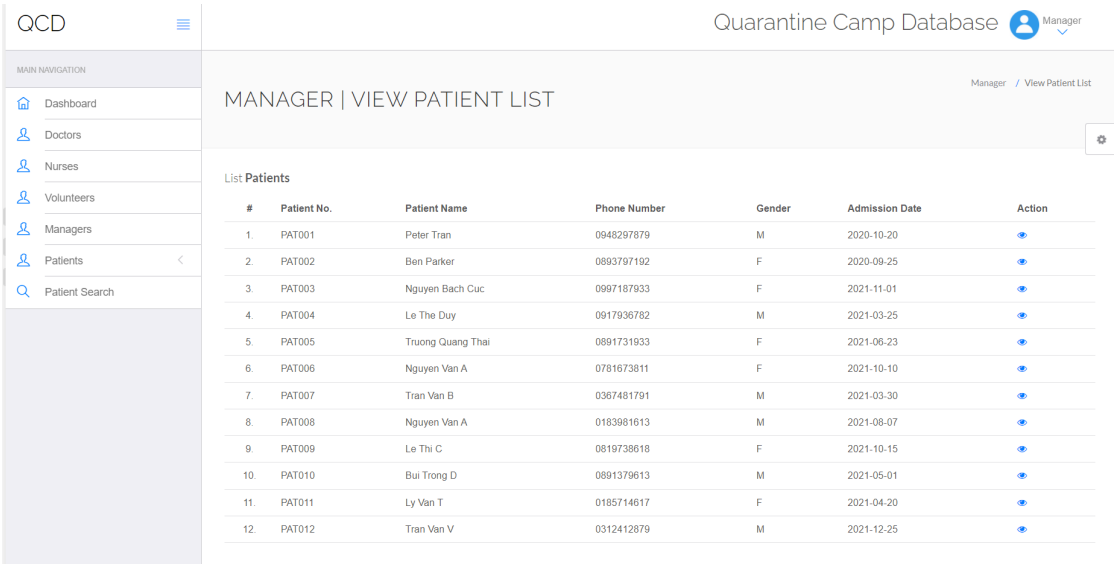


2.3.2.d List function

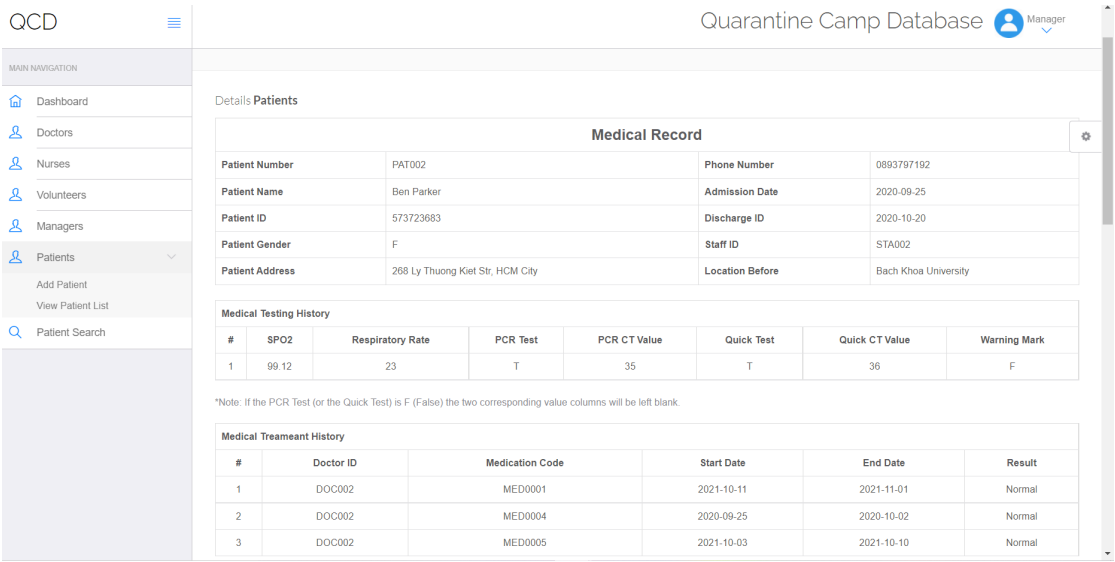
Requirement: List details of all testing which belong to a patient.

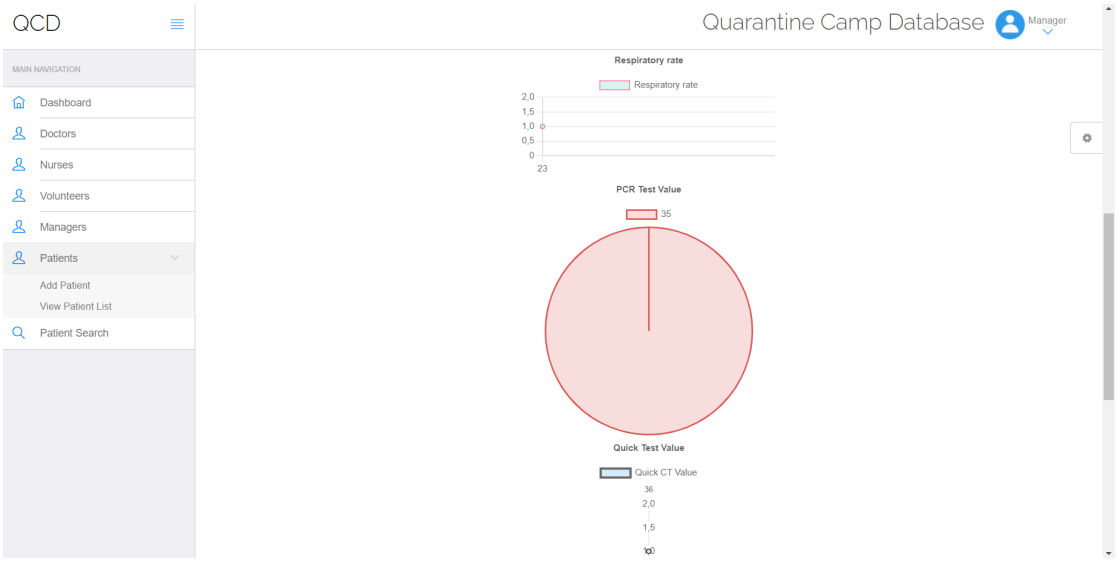
- Select the `Patient` line on the left toolbar, after appearing 2 command lines (`Add Patient` and `View Patient List`), you click on the `View Patient List` line, then we get the following interface.

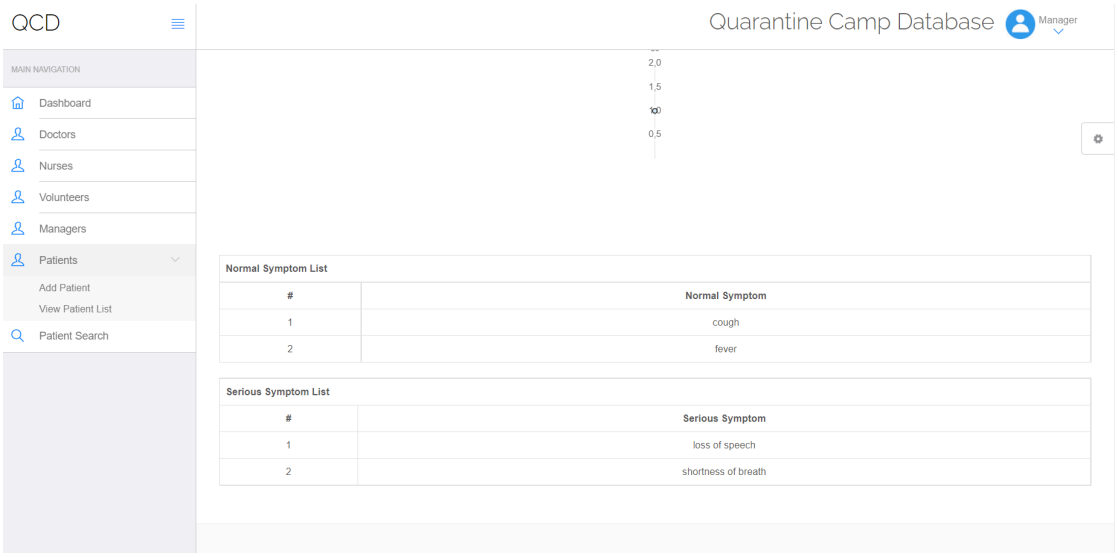




- To see the details of all testing of a patient, click on the "eye symbol" which is below the `Action` column on the right side. Then, you can see the result as pictures below which are the details of a patient whose name is "Ben Parker".







2.3.2.e Full detailed-report function

Requirement: Make a report that provides full information about the patient including demographic information, comorbidities, symptoms, testing, and treatment.