ADVANCED DATABASE

# Hospital Management System

# Team

Hagar Osama Mohammed

20221380675

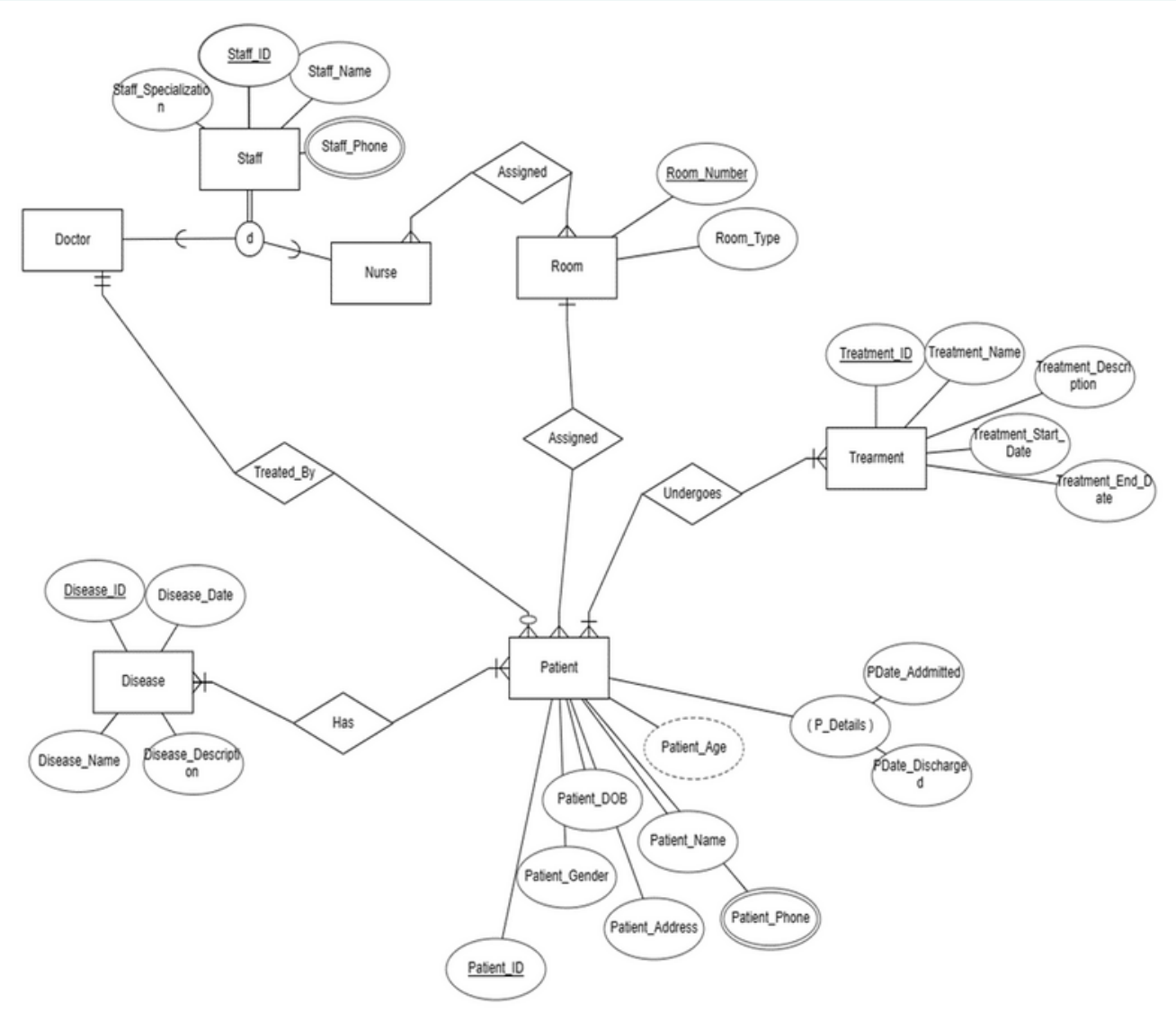
Yasmin Maher Ibrahim

20221379343

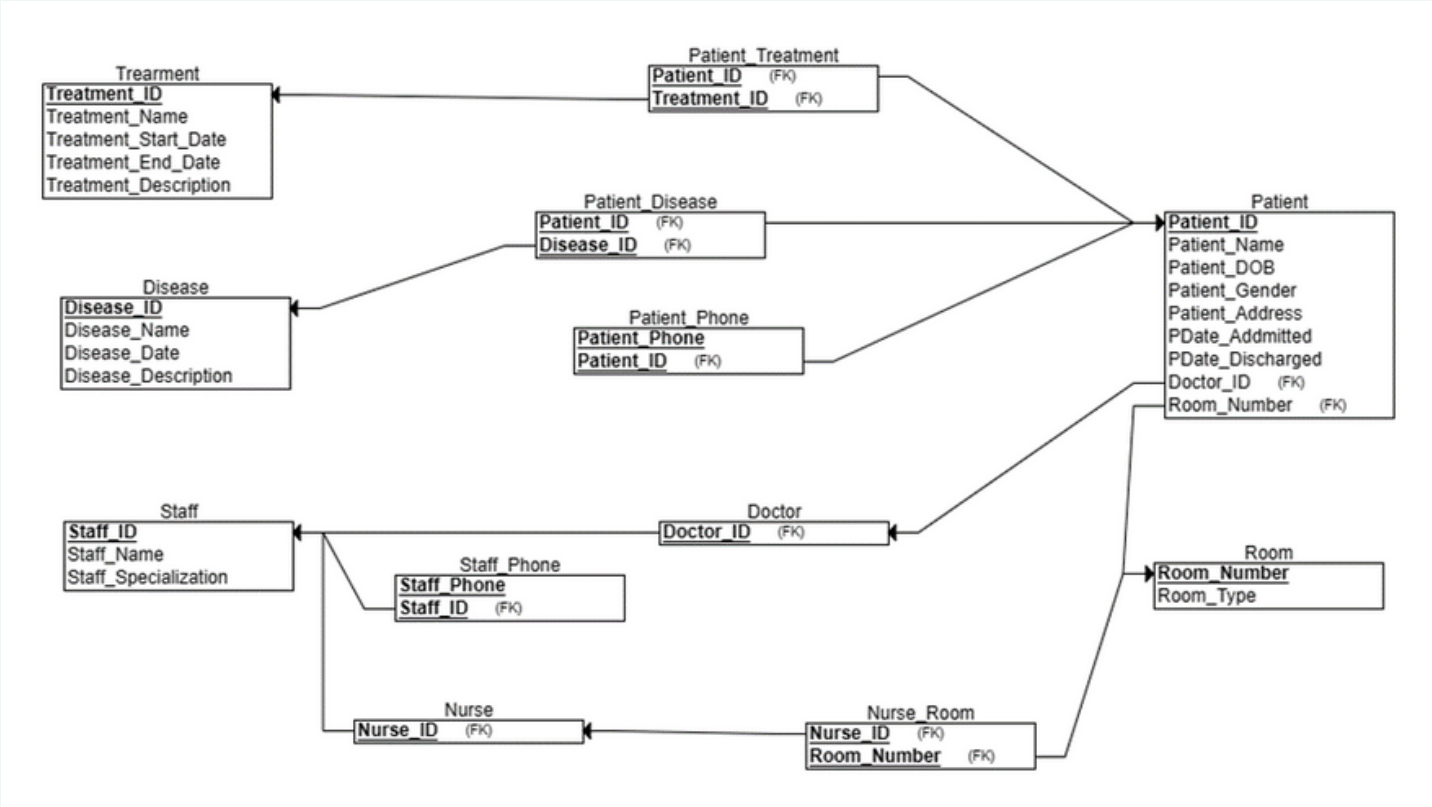
Mostafa Abdelaleem Nasr

20221459995

# ERD


Staff_ID
Staff_Name
Staff_Specializatio n
Staff
Staff_Phone
Assigned
Room_Number
Room_Type
Doctor
d
Nurse
Room
Treatment_ID
Treatment_Name
Treatment_Descri ption
Treatment_Start_ Date
Treatment_End_D ate
Trearment
Assigned
Treated_By
Undergoes
Disease_ID
Disease_Date
Disease
Has
Disease_Name
Disease_Descripti on
Patient
PDate_Addmitted
( P_Details )
PDate_Discharge d
Patient_Age
Patient_DOB
Patient_Name
Patient_Gender
Patient_Address
Patient_Phone
Patient_ID

# Relational Schema



# Logical Model

Trearment (Treatment_ID, Treatment_Name, Treatment_Start_Date, Treatment_End_Date, Treatment_Description)

Disease (Disease_ID, Disease_Name, Disease_Date, Disease_Description)

Room (Room_Number, Room_Type)

Staff (Staff_ID, Staff_Name, Staff_Specialization)

Nurse (Nurse_ID (FK -> Staff))

Doctor (Doctor_ID (FK -> Staff))

Nurse_Room (Nurse_ID (FK -> Nurse), Room_Number (FK -> Room))

Staff_Phone (Staff_Phone, Staff_ID (FK -> Staff))

Patient (Patient_ID, Patient_Name, Patient_DOB, Patient_Gender, Patient_Address, PDate_Addmitted, PDate_Discharged, Doctor_ID (FK -> Doctor), Room_Number (FK -> Room))

Patient_Treatment (Patient_ID (FK -> Patient), Treatment_ID (FK -> Trearment))

Patient_Disease (Patient_ID (FK -> Patient), Disease_ID (FK -> Disease))

Patient_Phone (Patient_Phone, Patient_ID (FK -> Patient))

# Normalization

## 1 NF

Staff

| Staff_ID | Staff_Name | Staff_Specialization | Staff_Phone |
|---|---|---|---|
| 11 | Emma Davis | Nurse | 123456789 |
| 12 | Oliver Wilson | Nurse | 111111111 |
| 13 | Sophia Taylor | Doctor | 222222222 |
| 14 | Mia Harris | Doctor | 333333333 |

Staff

| Staff_ID | Staff_Name | Staff_Specialization |
|---|---|---|
| 11 | Emma Davis | Nurse |
| 12 | Oliver Wilson | Nurse |
| 13 | Sophia Taylor | Doctor |
| 14 | Mia Harris | Doctor |

Staff_Phone

| Staff_ID | Staff_Phone |
|---|---|
| 11 | 123456789 |
| 12 | 111111111 |
| 13 | 222222222 |
| 14 | 333333333 |

- Each table has a primary key
- No multivalued attributes

# Normalization

## 2 NF

| Room_Number | Room_Type | Shift_Duration |
|---|---|---|
| 105 | General | 08:00-16:00 |
| 106 | ICU | 16:00-00:00 |
| 107 | General | 00:00-08:00 |
| 108 | ICU | 08:00-16:00 |

- It's in 1NF
- No partial functional dependencies, here Non-key attribute is fully dependent in the composite key (Room_Number, Room_Type)

# Normalization

## 3 NF

Patient_Before

| Patient_ID | Name | DOB | Gender | PDate_Addmitted | PDate_Discharged | Doctor_ID | Room_Number | Room_Type |
|---|---|---|---|---|---|---|---|---|
| 15 | Patient E | 2004-05-05 | Male | 2023-05-01 | 2023-05-31 | 13 | 105 | General |
| 16 | Patient F | 2005-06-06 | Female | 2023-06-01 | 2023-06-30 | 13 | 106 | ICU |
| 17 | Patient G | 2006-07-07 | Male | 2023-07-01 | 2023-07-31 | 14 | 107 | General |
| 18 | Patient H | 2007-08-08 | Female | 2023-08-01 | 2023-08-31 | 14 | 108 | ICU |

Patient

| Patient_ID | Name | DOB | Gender | PDate_Addmitted | PDate_Discharged | Doctor_ID | Room_Number |
|---|---|---|---|---|---|---|---|
| 15 | Patient E | 2004-05-05 | Male | 2023-05-01 | 2023-05-31 | 13 | 105 |
| 16 | Patient F | 2005-06-06 | Female | 2023-06-01 | 2023-06-30 | 13 | 106 |
| 17 | Patient G | 2006-07-07 | Male | 2023-07-01 | 2023-07-31 | 14 | 107 |
| 18 | Patient H | 2007-08-08 | Female | 2023-08-01 | 2023-08-31 | 14 | 108 |

Room

| Room_Number | Room_Type |
|---|---|
| 105 | General |
| 106 | ICU |
| 107 | General |
| 108 | ICU |

- It's in 1NF
- No partial functional dependencies

**Before Conversion to 3NF:**

Patient_Before (**Patient_ID** , Name,DOB , Gender , PDate_Addmitted , PDate_Discharged , Doctor_ID , Room_Number , Room_Type)

**Functional Dependencies:**

**Patient_ID** --> Name,DOB , Gender , PDate_Addmitted , PDate_Discharged, Doctor_ID , Room_Number , Room_Type

**Room_Number** --> Room_Type (**Transitive**)

**After Conversion to 3NF:**

Patient (**Patient_ID** , Name,DOB , Gender , PDate_Addmitted , PDate_Discharged , Doctor_ID , Room_Number)

**Functional Dependencies:**

**Patient_ID** --> Name , DOB , Gender , PDate_Addmitted , PDate_Discharged , Doctor_ID , Room_Number

Room (**Room_Number** , Room_Type)

**Functional Dependencies:**

**Room_Number** --> Room_Type

# Queries

```sql
ALTER TABLE Nurse_Room
ADD Shift_Duration VARCHAR(255) NOT NULL;
```

```sql
UPDATE Nurse_Room
SET Shift_Duration = '08:00-16:00'
WHERE Nurse_ID = 11 AND Room_Number = 105;

UPDATE Nurse_Room
SET Shift_Duration = '16:00-00:00'
WHERE Nurse_ID = 11 AND Room_Number = 106;

UPDATE Nurse_Room
SET Shift_Duration = '00:00-08:00'
WHERE Nurse_ID = 12 AND Room_Number = 107;

UPDATE Nurse_Room
SET Shift_Duration = '08:00-16:00'
WHERE Nurse_ID = 12 AND Room_Number = 108;
```

| Nurse_ID | Room_Number | Shift_Duration |
|---|---|---|
| 11 | 105 | 08:00-16:00 |
| 11 | 106 | 16:00-00:00 |
| 12 | 107 | 00:00-08:00 |
| 12 | 108 | 08:00-16:00 |

# Query Optimization

## Query 1

```sql
SELECT P.Patient_Name, T.Treatment_Name
FROM Patient P
JOIN Patient_Treatment PT ON P.Patient_ID = PT.Patient_ID
JOIN Trearment T ON PT.Treatment_ID = T.Treatment_ID;
```

| Patient_Name | Treatment_Name |
|---|---|
| Patient E | Psychotherapy |
| Patient F | Radiation Therapy |
| Patient G | Radiation Therapy |
| Patient H | Stem Cell Transplant |

**JOIN** operations are more efficient than subqueries.

# Query Optimization

## Query 2

```
SELECT S.Staff_Name, COUNT(P.Patient_ID) as Patient_Count
FROM Staff S
JOIN Doctor D ON S.Staff_ID = D.Doctor_ID
JOIN Patient P ON D.Doctor_ID = P.Doctor_ID
GROUP BY S.Staff_Name;
```

| Staff_Name | Patient_Count |
|---|---|
| Mia Harris | 2 |
| Sophia Taylor | 2 |

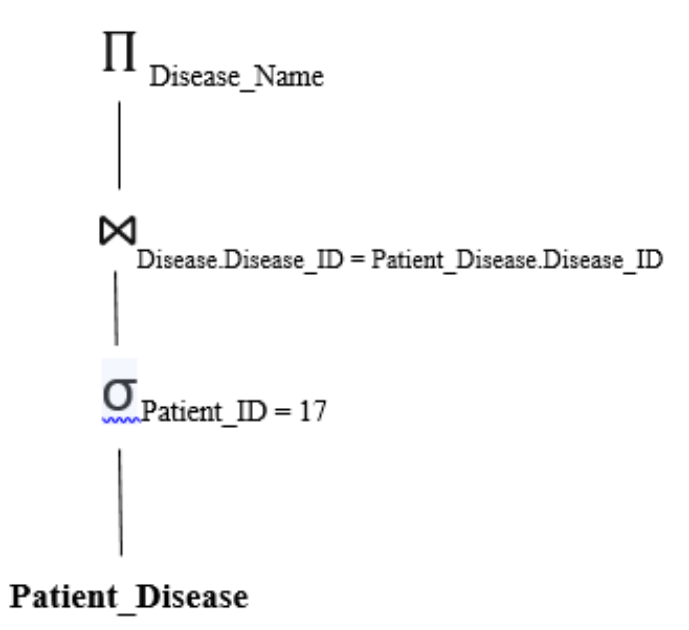
**GROUP BY,** aggregates function (**COUNT**) handled as a final step.

# Query Optimization

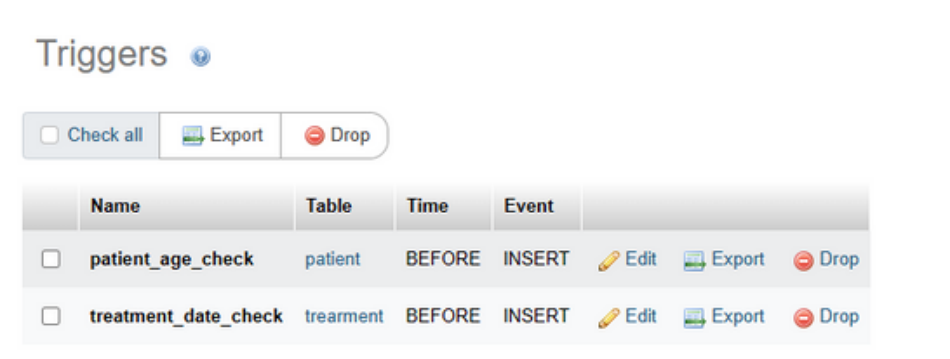
## Query 3

```sql
SELECT Disease.Disease_Name
FROM Disease
INNER JOIN Patient_Disease ON Disease.Disease_ID = Patient_Disease.Disease_ID
WHERE Patient_Disease.Patient_ID = 17;
```

### It's Relational Algebra



# Triggers




```
1 BEGIN
2     IF NEW.Patient_DOB = 0 THEN
3         SIGNAL SQLSTATE '45000'
4         SET MESSAGE_TEXT = 'Age cannot be zero';
5     END IF;
6 END
```

```
1 BEGIN
2     IF NEW.Treatment_End_Date < NEW.Treatment_Start_Date THEN
3         SIGNAL SQLSTATE '45000'
4         SET MESSAGE_TEXT = 'Treatment End Date cannot be before
  Start Date';
5     END IF;
6 END
```

“The main purpose of triggers here, is to alert the user if he entered a wrong input.”

# Index

```
1  CREATE INDEX Patientin ON Patient (Patient_id) USING HASH;
2
3
```

```
1  CREATE INDEX Doctoridx ON doctor (Doctor_ID) USING HASH;
2
3
```

```
1  CREATE INDEX nurseidx ON nurse (Nurse_ID) USING HASH;
2
3
```

```
1  CREATE INDEX treatmentidx ON trearment (Treatment_ID) USING HASH;
2
3
```

“As a part of optimization, we created an index to fast search, insert, delete, and update processes.”

# PHP Code

## Connect to the database

```php
<?php
$host = 'localhost';
$username = 'root';
$password = '';
$dbname = 'hospital';


$conn = new mysqli($host, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

## Get the data from both patient and disease tables by using join query

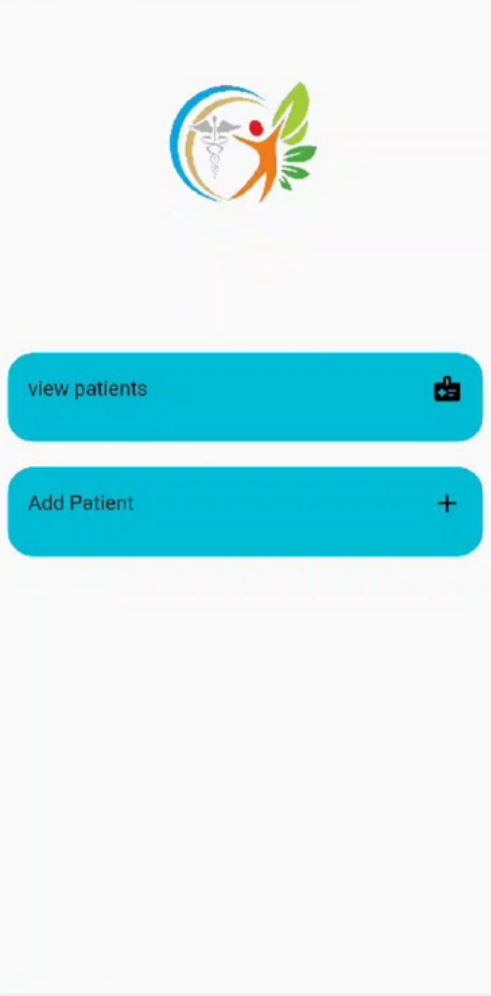
```php
<?php
include("connection.php");
$q = "SELECT p.Patient_ID ,Patient_Name, Patient_Gender, Patient_Address , Disease_Name
  FROM patient p  JOIN
 disease d WHERE d.Disease_ID = (SELECT Disease_ID FROM patient_disease WHERE Patient_ID=p.Patient_ID)";
$customer_pre = $conn->query($q);
$customers = array();

while ($row = $customer_pre->fetch_assoc()) {
    $customers[] = $row;
}

$response = json_encode($customers);

if (count($customers) == 0) {
    http_response_code(404);
} else {
    http_response_code(200);
}
echo $response;
```

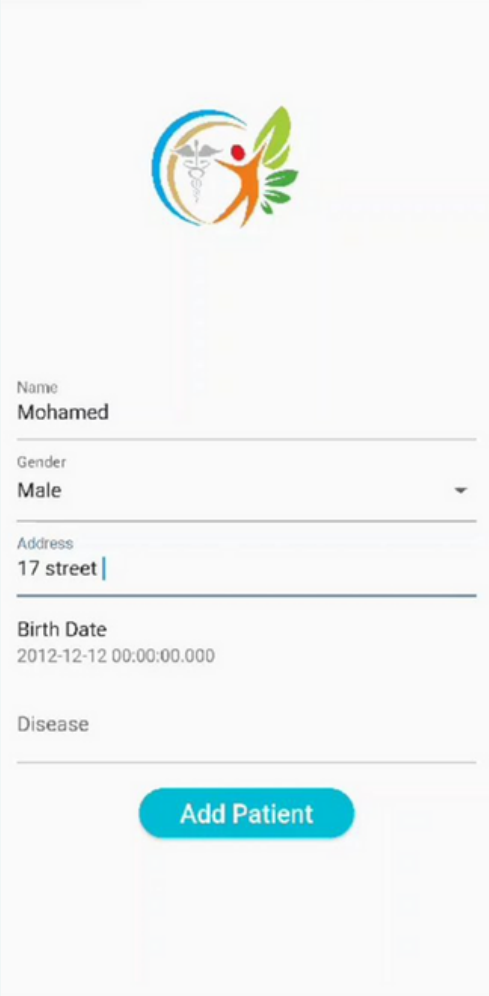
# Add new patient using post to receive the data from the GUI

```php
<?php

include("connection.php");


$name = $_POST["name"];
$gender = $_POST["gender"];
$address = $_POST["address"];
$birth_date = $_POST["birth_date"];

$disease = $_POST["disease"];


$birth = date('Y-m-d', strtotime($birth_date));


$insertPatient = "INSERT INTO `patient`( `Patient_Name`, `Patient_Gender`,
`Patient_Address`,`Patient_DOB`,`Doctor_ID`,`Room_Number`)
VALUES ('$name','$gender','$address','$birth',13,105)";

$insertPatientDisease = "INSERT INTO `patient_disease`(`Patient_ID`, `Disease_ID`)
VALUES ((SELECT MAX(Patient_ID) FROM patient),
(SELECT Disease_ID FROM disease WHERE Disease_Name = '$disease'));";


$prepare = $con->prepare($insertPatient);
$prepare->execute();
$count = $prepare->rowCount();


$prepare = $con->prepare($insertPatientDisease);
$prepare->execute();
if ($count == 0) {
    http_response_code(404);
} else {
    http_response_code(200);
}
echo $count;
```

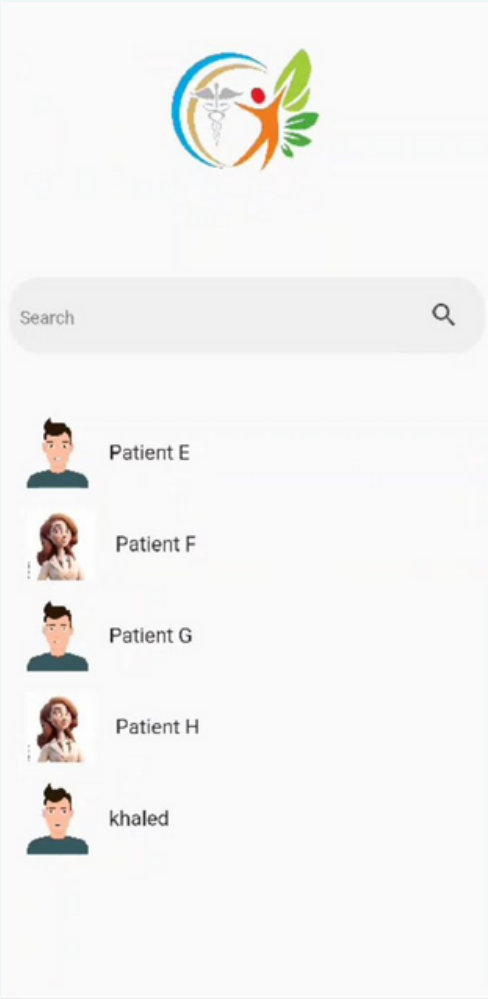
# GUI

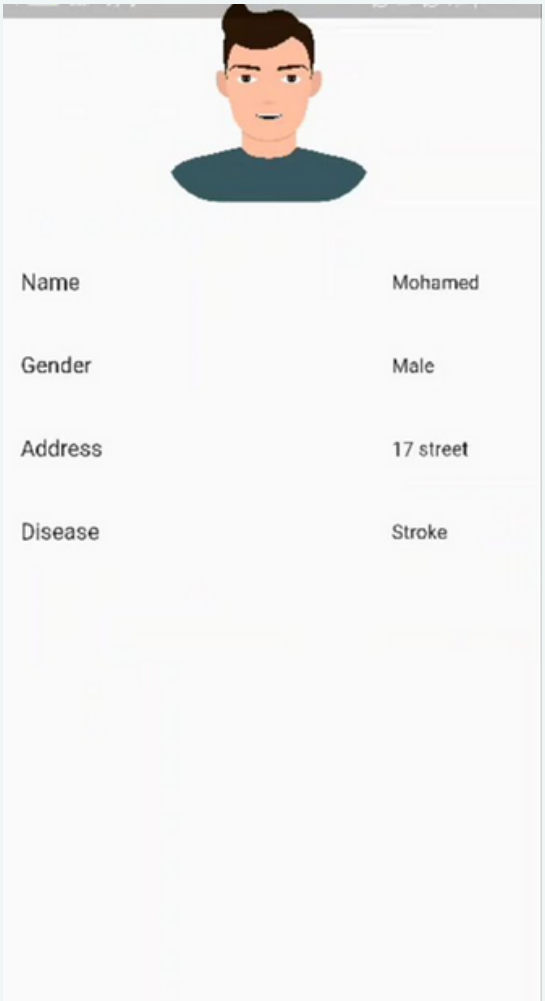
## Some screens of  Mobile GUI



# GUI


Search
Patient E
Patient F
Patient G
Patient H
khaled


Name
Mohamed
Gender
Male
Address
17 street
Disease
Stroke