# Technical Documentation

# Introduction

This document is a technical documentation about Aboody primary care clinic in Al Khalidi Hospital and other outside branches. It includes the final structure of the database, with the database developments processes, user interface, with high quality maintenance. And it will include testing. And an evaluation of the solution on database.

# Physical Schema

Graphical user interface, application

Description automatically generated

# Database Development

## Database Overview

|  |  |  |
| --- | --- | --- |
| **Table** | **Name** | **Description** |
|  | Doctor | Doctor table, contain these information for doctor: id, Doctor first and second name, speciality of the doctor, email ,salary, doctor birthdate, username and password and branch number which doctor work in |
|  | Doctor\_Phone | This table contains doctor phones |
|  | Doctor\_patient | This table include patients with doctors which they take care of them |
|  | Doctor-address | This table include addresses of doctors: their city, street name and building number( their homes locations) |
|  | Branch | This table include branch information: it’s number, phone, and id of employees and doctors which work in |
|  | Branch-address | This table includes information about branches location(city, the name of the street which the branch is in, number of building also) |
|  | Branch\_name-and-opening\_date | This table includes information about branches name and their opening dates |
|  | Employee | This table includes information about the different employees inside branches and their role and it’s includes their names also |
|  | Prescription | This table includes information about prescription: time and date of prescription, its number and which doctor wrote it and for which patient. |
|  | Prescription\_medication | This table includes information about what Medications doctors have given patients in the prescription: Medicine name and explain of use of the medicine. |
|  | Patient | This table includes information about patients: their ids first and last names, their emails and other necessary informations. |
|  | Patient\_address | This table includes information about patients’ homes locations, the city, the street name and the number of building patient has of each of his/her location |

|  |  |  |
| --- | --- | --- |
| **View** | **Name** | **Description** |
|  | Show\_patient\_num | This is to count the patients numbers in all branches |
|  | Show\_doctor\_num\_manager\_1111 | view so manager with (1111) id can show doctors number at his branch only |
|  | Show\_doctor\_num\_manager\_1234 | view so manager with (1234) id can show doctors number at his branch only |
|  | Show\_doctor\_num\_manager\_1342 | view so manager with (1342) id can show doctors number at his branch only |
|  | Show\_doctor\_num\_manager\_3344 | view so manager with (3344) id can show doctors number at his branch only |
|  | Show\_doctor\_num\_manager\_3344\_2 | view so manager with (3344) id can show doctors number at his branch only(he has more than one branch) |

|  |  |  |
| --- | --- | --- |
| **Procedure** | **Name** | **Description** |
|  | addBranch | This branch to add branch to branch table |
|  | addPatient | This procedure is to add patients to patient table |
|  | Add\_prescription\_to\_patient | This to add a prescription to the patient |
|  | deleteBranch | This to delete desire branches |
|  | delete\_doctor\_manager\_1111 | Procedures so manager with (1111) id can delete doctors at his branch only |
|  | delete\_doctor\_manager\_1234 | Procedures so manager with (1234) id can delete doctors at his branch only |
|  | delete\_doctor\_manager\_1342 | Procedures so manager with (1342) id can delete doctors at his branch only |
|  | delete\_doctor\_manager\_3344 | Procedures so manager with (3344) id can delete doctors at his branch only |
|  | delete\_doctor\_manager\_3344\_2 | Procedures so manager with (3344) id can delete doctors at his branch only(the other branch)(he is the manager for two branches) |
|  | Delete\_manager | This is to delete managers from branch |
|  | Delete patient | This is to delete patient at any branch |
|  | Delete\_prescription\_to\_patient | This is to delete patients prescription |
|  | doctor\_show\_his\_information | This is to allow doctor see his information |
|  | Doctor\_update\_his\_information | This is to allow doctor update his information |
|  | INSERT\_doctor\_manager\_1111 | Procedures so manager with (1111) id can add doctors at his branch only |
|  | INSERT\_doctor\_manager\_1234 | Procedures so manager with (1234) id can add doctors at his branch only |
|  | INSERT\_doctor\_manager\_1342 | Procedures so manager with (1342) id can add doctors at his branch only |
|  | INSERT\_doctor\_manager\_3344 | Procedures so manager with (3344) id can add doctors at his branch only |
|  | INSERT\_doctor\_manager\_3344\_2 | Procedures so manager with (3344) id can add doctors at his branch only(the other branch)(he is the manager for two branches) |
|  | Insert\_manager | This is to insert |
|  | Patient\_show\_his\_information | This is enable patient to show his information |
|  | Patient\_show\_his\_prescription | This is enable patient to show his prescription |
|  | Patient\_update\_his\_information | This is enable patient to update his information |
|  | ShowDoctorNum | This is to show doctor number at any branch |
|  | showHealthRecord | This is to show health record to any patient |
|  | showSalaryOfDoctors | This is to show salary of doctors at any branch |
|  | Show\_salaries\_by\_manager | This is to show salary by manager only for his branch |
|  | Update\_doctor\_manager\_1111 | Procedures so manager with (1111) id can update doctors at his branch only |
|  | Update\_doctor\_manager\_1234 | Procedures so manager with (1234) id can update doctors at his branch only |
|  | Update\_doctor\_manager\_1342 | Procedures so manager with (1342) id can update doctors at his branch only |
|  | Update\_doctor\_manager\_3344 | Procedures so manager with (3344) id can update doctors at his branch only |
|  | Update\_doctor\_manager\_3344\_2 | Procedures so manager with (3344) id can update doctors at his branch only (the other branch)(he is the manager for two branches) |

## Security

|  |  |  |  |
| --- | --- | --- | --- |
| **User name** | **Privilege Command** | **Description** | **Screenshot** |
| doctor | |  |  |  | | --- | --- | --- | | |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`patient\_show\_his\_prescription` TO 'doctor'@'localhost' | |  | | |  | | We give permission to user to execute procedure which allow to doctor to show patient prescription |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'doctor'@'localhost' | |  | | We give permission to user to execute procedure which allow to doctor to show patient health record |  |
| |  |  |  | | --- | --- | --- | | |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`doctor\_update\_his\_information` TO 'doctor'@'localhost' | |  | | |  | | We give permission to user to execute procedure which allow to doctor to update his information |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`doctor\_show\_his\_information` TO 'doctor'@'localhost' | |  | | We give permission to user to execute procedure which allow to doctor to showhis information |  |
| GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`patient\_show\_his\_information` TO 'doctor'@'localhost' | We give permission to user to execute procedure which allow to doctor to show patient information |  |
| patient | |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`patient\_update\_his\_information` TO 'patient'@'localhost' | |  | | We give permission to user to execute procedure which allow to patient to update his information |  |
|  | |  | | --- | |  | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`patient\_show\_his\_prescription` TO 'patient'@'localhost' | |  | | We give permission to user to execute procedure which allow to patient to show his prescription |  |
|  | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`patient\_show\_his\_information` TO 'patient'@'localhost' | We give permission to user to execute procedure which allow to patient to show his information |  |
| CEO | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_patient\_num` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to show patient numbers in each branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showsalaryofdoctors` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to show doctor salaries at each branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`addbranch` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to add a branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_manager` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to delete a manager from a branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showdoctornum` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to show doctor number at each branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_patient` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to delete patient |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`insert\_manager` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to insert manager to a branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`addpatient` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to add patient to a branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'CEO'@'localhost' | |  | | We give permission to user to execute procedure which allow to CEO to show health record for each patient |  |
| GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`deletebranch` TO 'CEO'@'localhost' | We give permission to user to execute procedure which allow to CEO to delete a branch |  |
| Manager1111 | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_doctor\_num\_manager\_1111` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to show doctor numbers at his branch |  |
|  | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_patient\_num` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to show patient numbers |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_prescription\_to\_patient` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to delete prescription |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`update\_doctor\_manager\_1111` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to update doctor information |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`insert\_doctor\_manager\_1111` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 insert doctor to a branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`add\_prescription\_to\_patient` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to add prescription to a patient |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to show health record for patient |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showsalaryofdoctors` TO 'Manager1111'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1111 to show doctor salaries at each branch |  |
| GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_doctor\_manager\_1111` TO 'Manager1111'@'localhost' | We give permission to user to execute procedure which allow to Manager1111 to delete from a branch |  |
| Manager1234 |  |  |  |
|  | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_doctor\_num\_manager\_1234` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to show doctor numbers only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_doctor\_manager\_1234` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to delete doctor from his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`update\_doctor\_manager\_1234` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to update doctor information only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`add\_prescription\_to\_patient` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to add prescription to patient |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to show health record for each patient |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`insert\_doctor\_manager\_1234` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to omsert doctor only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_prescription\_to\_patient` TO 'Manager1234'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1234 to delete prescriptions from patients |  |
| GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showsalaryofdoctors` TO 'Manager1234'@'localhost' | We give permission to user to execute procedure which allow to Manager1234 to show doctor salaries for each branch |  |
| Manager1342 | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_doctor\_num\_manager\_1342` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to show doctor numbers only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_prescription\_to\_patient` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to delete doctor only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`add\_prescription\_to\_patient` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to add prescriptions for patients |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_doctor\_manager\_1342` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to delete doctor only for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`update\_doctor\_manager\_1342` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to update doctor information for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to show patients health record |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showsalaryofdoctors` TO 'Manager1342'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager1342 to show salaries of doctors |  |
| GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`insert\_doctor\_manager\_1342` TO 'Manager1342'@'localhost' | We give permission to user to execute procedure which allow to Manager1342 to insert doctor for his branch |  |
| Manager3344 | |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_doctor\_num\_manager\_3344\_2` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add show doctor number at his second branch |  |
| |  | | --- | | GRANT SELECT ON `primary\_care\_clinic`.`show\_doctor\_num\_manager\_3344` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add show doctor number at his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showsalaryofdoctors` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add show doctor salaries |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`add\_prescription\_to\_patient` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add prescription for patients |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`showhealthrecord` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add show health record for patients |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_prescription\_to\_patient` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to delete prescription for patientes |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`insert\_doctor\_manager\_3344` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add doctor for his branch |  |
| |  | | --- | | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`delete\_doctor\_manager\_3344` TO 'Manager3344'@'localhost' | |  | | We give permission to user to execute procedure which allow to Manager3344 to add delete doctor at his branch |  |
|  | GRANT EXECUTE ON PROCEDURE `primary\_care\_clinic`.`update\_doctor\_manager\_3344` TO 'Manager3344'@'localhost' | We give permission to user to execute procedure which allow to Manager3344 to update doctor information at his branch |  |

## User Interface

### Flowchart and Data Movement Diagrams

* FlowChart

A picture containing sketch, diagram, drawing, plan

Description automatically generated

DFD level zero

A picture containing line, diagram, screenshot, plot

Description automatically generated

DFD level one

A picture containing diagram, technical drawing, sketch, plan

Description automatically generated

### Interfaces Development

|  |  |  |  |
| --- | --- | --- | --- |
| **Page ID** | **Title** | **Description** | **Screenshot** |
|  | Doctor | In this page user can view, update, delete doctor information. He can add doctor, delete doctor. |  |
|  | Doctor Address | In this page, user can update doctor address(city, street name, building number). He can add/delete/view doctor address |  |
|  | Doctor Patient | In this page, user can add/delete/update/view doctor and patient related to each other and date and time. |  |
|  | Doctor Phone | In this page, user can add/update/delete/view doctor phones |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Type** | **Description** | **screenshot** |
|  | All cases of PK | I want to test that the primary key does not accept null |  |
| In this test, I want to insert two equal values to verify that primary key does not accept any value that is not unique |  |
|  | All cases of | Test that both will update (primary key, and foreign key) if I change The primary key that related to the foreign key(on update cascade) |  |
| I want to check if both are deleted(primary key, and foreign key) when I delete the primary key releated to the foreign key  (On delete Cascade) |  |
|  | Unique | I want to check that the attributes with unique constrains does not accept any not unique values |  |
|  | Default | I don’t have default constrains |  |
|  | Not null | I want to check that the attributes with constrain Not null does not accept ‘Null’ value |  |
|  | Check | I don’t have check |  |
|  | ……… |  |  |

# Maintenance

## Database recovery & backups

In database:

We can recovery and backups: to take a backup of your database, so, if something goes wrong to the database or if your operating systems goes down or anything a backup is taken to the database, in order to be able to recover database.

To take a backup from database you have to make export to the database:

* From phpMyAdmin, we choose the database we want to make backup(export) for it:

A screenshot of a computer

Description automatically generated

* Then we goes to the ‘Export’ column:

A screenshot of a computer

Description automatically generated with low confidence

* Then we press on Go button in order to take a backup and the backup file will immediately start downloading after pressing on Go Button:

A screenshot of a computer

Description automatically generated with low confidence

* Done:

A picture containing text, screenshot, font, rectangle

Description automatically generated

The downloaded file

To make recovery to the database, so we must now import the backup file to a database as the following:

* First, we create a new database to import the backup file into.

A screenshot of a computer

Description automatically generated with low confidence

* Then we press on the database:

A picture containing text, font, screenshot

Description automatically generated

* Go to the import column:

A screenshot of a computer

Description automatically generated with medium confidence

* We press choose file button to choose the file we want to import:

A screenshot of a computer

Description automatically generated with medium confidence

* Now choose the file you want to import and press open:

A screenshot of a computer

Description automatically generated

* Finally, press Go button in order to complete the import process:

A screenshot of a computer

Description automatically generated with low confidence

* Done, and now all the table and details has been recovered:

A screenshot of a computer

Description automatically generated

## Database maintenance in general

In general, there are different issues must be taken into consideration when we talk about maintenance of database:

So, we must take care about backup and recovery. To take a backup from the database each period of time so we can recover database data when something cause damage to it. And we must take care about monitor the database performance and tuning, so, identify bottlenecks, we optimeize the usage of resource, and ensure efficient execution of query. Also, cleaning database from garbage, which it is the old unnecessary data, and data out of date and redundant data. So this can reduce the size of database and improve its performance. Also, security and Access control must be takin into consideration, regularly review user access, force password policy and do the necessary security updates. Also, capacity planning, This is must be the responsibility of DBA to monitor the growth of database over time and make it to estimate the requirements of storage in the future. Also Documentation, up to date documentation is essential for database maintenance. These things and more must be taken into consideration in terms of maintenance database.

# Testing

## Data Validation

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Query Description | **Screenshot (query + result)** | Result validation |
| 1. | This is query count number of prescript |  |  |
| 2. | This is query to add patient |  |  |
| 3. | This is query to add branch |  |  |
| 4. | This is query to insert doctor |  |  |

## Output Validation

## Security Validation

**Note**: you need to test the given and not given privileges.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **User Name** | **Description of privilege/no privilege** | **Screenshot (query + result)** |
|  | CEO | Add branch  Delete branch  Patient number show  Add manager to a branch  Delete manager  Show doctor salaries  Show doctor numbers  Show patient healthrecord  AddPatient  Delete patient |  |
|  | patient | Show his information  Update his information  Show his prescription |  |
|  | doctor | Show patient health record  Show his information  Update his information |  |
|  | Manager1111 |  |  |
|  | Manager1234 |  |  |
|  | Manager1342 |  |  |
|  | Manager3344 |  |  |

### 

## GUI Validation

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **screenshot** |
|  | In Doctor page, user(root) can add/delete/view/update doctor information. |  |
|  | In Doctor Address page, user(root) can add/delete/view/update doctor addresses information. |  |
|  | Doctor Patient page is to add doctor and patient together and date and time. And also to update, view and delete them. |  |
|  | In Doctor Address page, user(root) can add/delete/view/update doctor phones information. |  |

## Assess whether meaningful data has been extracted

Recording to that (if meaningful data has been extracted or not). The database that I have designed is for primary care clinic.

The extracted data from the database is extracted, since:

1) The data extracted is related to the the scenario requirements, and return a reliable data which can contribute to the desired outcomes of a system of that type.

2) The extracted data is useful, since it can used to informed decision making.

3) The data extracted includes patient records, which are comprehensive, whatever it was, personal data, medical history, medications. But there is no diagnoses which must be added later

4) The extracted data includes information about appointments, such as time, date, medication, patient details. The data extracted is relevant and complete. It covers this functionality.

5) The extracted data includes information about employees, doctors, patients, and other data related to the scenario. All data are relevant and complete and cover the functionalities. But there are some limitations such as the addition of manager is good, but not flexible as its expected.

6) The extracted data is consistent with the required information. So, whatever user insert or extract, the information is consistence. So, no wrong information or interrelated data can be extracted.

Based on this assessment, the data has been extracted is meaningful. But there should be some changes and improvements to reach a better results such as improve addition manager functionality.

### Assess the effectiveness of testing

1) the test has cover all things I have done, procedures, views, quieres, and interfaces.

2) the test has detected many bugs and uncovering issues.

3) The test has not taken long time, it has been done in a very suitable time.

4) The test has been documented including clear test cases and test results, and the documentation accurately captured the testing outcomes, and it can be easily understood by other stakeholders and followed.

# Evaluation of database solution

## Effectiveness of the database solution based on user and system requirement

## Suggested improvements

## Evaluation based on improvements needed