# Technical Documentation

# Introduction

I was approached by Al Basheer Hospital to build a database for their Department of Ophthalmology.

The Hospital asked for a system that could store and manage patient information, ophthalmologist information, eye examination information, treatment information, and emergency contact information, this information was collected from the management team and the ophthalmology department.

# Physical Schema



**Note:** the relation between the patient table and the ophthalmologist table is one to one but the SQL physical designer have some issues

# Database Development

## Database Overview

|  |  |  |
| --- | --- | --- |
| **Table** | **Name** | **Description** |
|  | Patient | This table stores the personal and insurance details of all patients. Fields include Patient\_ID, Insurance\_ID, first name, last name, phone number, date of birth, username, password, and insurance degree. |
|  | Ophthalmologist | This table contains the professional and contact details of all ophthalmologists. include Ophthalmologist\_Id, first name, last name, phone number, specialty, username, and password. |
|  | Ophthalmologist-Patient | This table acts as a bridge table linking patients to their respective ophthalmologists. include Ophthalmologist\_Id and Patient\_Id. |
|  | Eye-examination | This table holds information about the eye examinations conducted for patients by ophthalmologists. include Eye examination\_id, Patient\_Id, Ophthalmologist\_Id, examination date, type of examination, and results of the examination. |
|  | Treatment | This table stores information on the treatments prescribed to patients by ophthalmologists. include Treatment\_Id, Patient\_Id, Ophthalmologist\_Id, dosage, frequency, follow-up appointments, and treatment type. |
|  | Work schedule | This table holds the work schedule of ophthalmologists. include Ophthalmologist\_Id, day, and work schedule. |
|  | Patient-Medical history | This table contains the medical history of patients. include Patient\_Id and Medical history. |
|  | Insurance | This table provides details on different insurance companies. Fields include Insurance\_Id and Insurance name. |
|  | Emergence contact | This table holds the emergency contact details for patients. include Contact\_Id, Patient\_Id, first name, last name, relationship, and phone number. |
|  | Ophthalmologist-Medical degree | This table stores the medical degree information of ophthalmologists. include Ophthalmologist\_Id and Medical degree. |

|  |  |  |
| --- | --- | --- |
| **View** | **Name** | **Description** |
|  | Patient\_View | This virtual view table has some information about each patient and this information are gathered from multiple tables which are the patient table and insurance table and patient medical history table and treatment table.  This table was made in case of the patient user or the ophthalmologist user want to see any information about any patient.  The information in this virtual table are: first name, last name, phone number, username, password, date of birth, insurance name, and insurance degree, emergency contact first name, emergency contact last name, emergency contact phone number, emergency contact relationship, patient medical history and the follow up appointment. |
|  | Ophthalmologist\_View | This virtual view table has some information about each ophthalmologist and this information are gathered from multiple tables which are the ophthalmologist table and the work schedule table.  This table was made in case of the ophthalmologist user wants to see any information about any ophthalmologist.  The information in this virtual table are: first name, last name, phone number, username, password, specialty, and the work schedule |
|  | Insurance\_Companies | This virtual view table show the names of all the insurance companies that are available in this hospital and these names are from the insurance table  This table was made in case any of the patients or the ophthalmologists want to know what insurance are available in this hospital |

**Benefits of creating a virtual view table:**

A view makes querying data easier by preventing the need to repeatedly connect different tables. With only one SELECT command, you can get data from several tables. (headway, 2022), (Wadje, 2022)

Even if the fundamental information changes, views may still be able to present the data in a consistent way. After a database repair, this may be very helpful for keeping backward compatibility with previous systems. (headway, 2022), (Wadje, 2022)

Views can give a more effective approach to retrieving data for complex computations. The database can figure out the view and save the results for later use rather than processing the data every time a request is made. (headway, 2022), (Wadje, 2022)

It offers an abstraction level. The end users do not need to be aware of the complicated relationships among the tables, the specifics of the table structures, or any other aspect of the database design. Through a straightforward display, they may easily engage with the data. (headway, 2022), (Wadje, 2022)

headway, headway (2022) *Organize complex query - advantages of views in SQL*, *RSS*. Available at: https://www.headway.io/blog/advantages-of-sql-views (Accessed: 20 June 2023).

Wadje, V. (2022) *Advantages and disadvantages of views in SQL server*, *C# Corner*. Available at: https://www.c-sharpcorner.com/blogs/advantages-and-disadvantages-of-views-in-sql-server1 (Accessed: 20 June 2023).

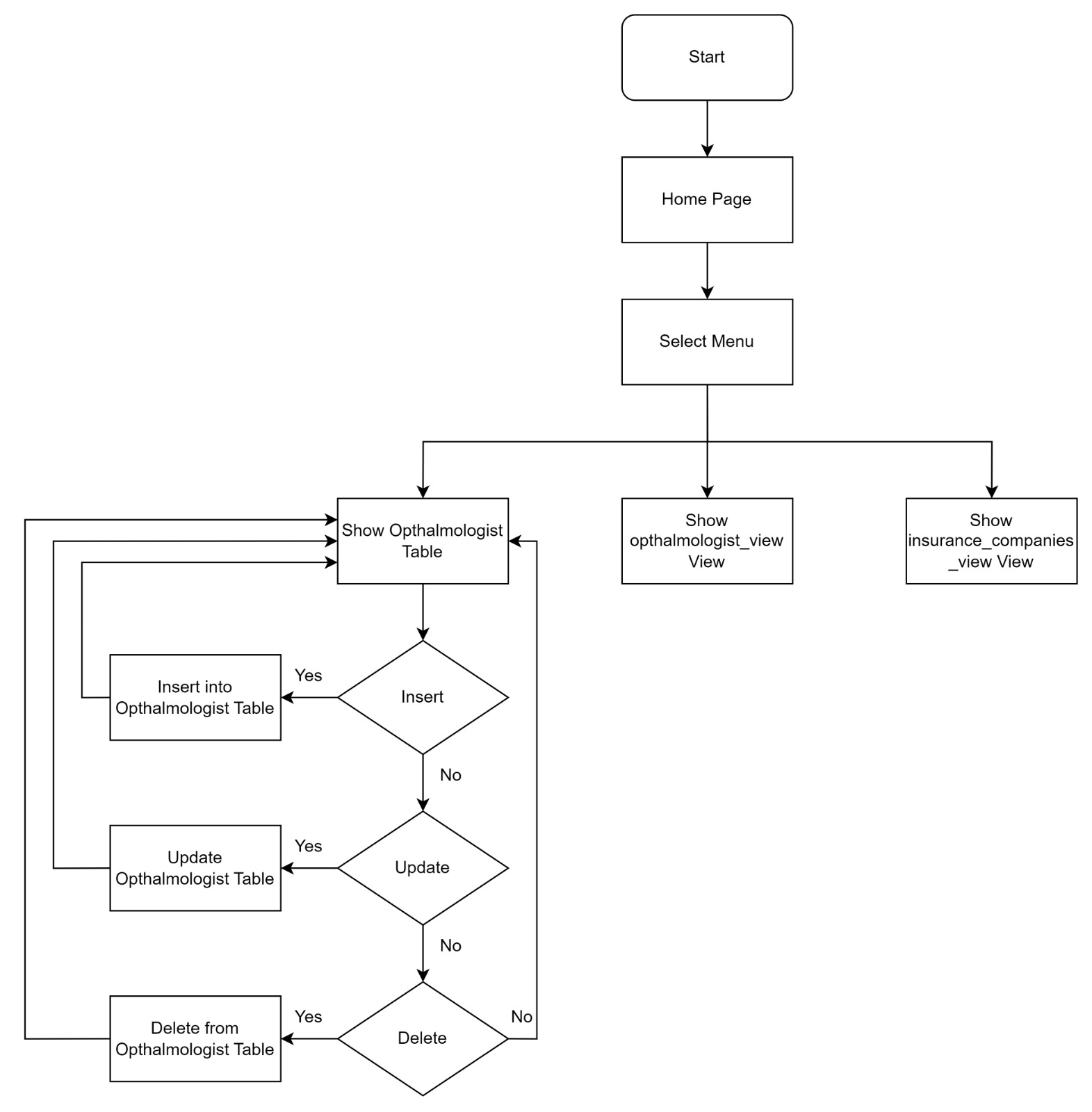
|  |  |  |
| --- | --- | --- |
| **Procedure** | **Name** | **Description** |
|  | InsertPatient | This procedure takes all the attribute which are in the patient table as a parameter.  This procedure (function) was made in case we want to insert a new patient to the patient table in the database just by calling this procedure and giving it all the values of the parameters. |
|  | UpdatePatientMedicalHistory | This procedure takes two parameters which are the patients id (so this procedure knows which patient we want to update his medical history in the table patient medical history), the second parameter is the new medical history.  So as mentioned this procedure (function) was made in case we want to change any of the patient’s medical history in the patient medical history table with an easy step just by calling the procedure and giving it the patient id and the new medical history. |
|  | UpdatePatientEmergencyContact | This procedure takes five parameters which are all the attributes which are in the emergency contact table.  This procedure (function) was made in case we want to update the patient emergence contact information in the emergency contact table in the database just by calling this procedure and giving it all the values of the parameters. |
|  | UpdateFollowUpAppointment | This procedure takes two parameters which are the patient id (to know which patient we want to change his following appointment) and the second parameter is the updated follow up appointment.  This procedure was made in case of any changes on the follow up appointment in the treatment table just by calling this function. |
|  | DeletePatient | This procedure takes just one parameter which is the patient id.  This procedure was made in case if we want to delete a patient from the patient table. |
|  | GetPatientDetails | This procedure takes two parameters which are the patient username and password and returns details about the patient.  This procedure was made if we want to see all the information that are related just for one patient.  **Note:** each patient has a username and a password so this function just return the information of the patient that his username and password matches the inserted username and password parameters. |
|  | InsertOphthalmologist | This procedure takes all the attribute which are in the ophthalmologist table as a parameter.  This procedure (function) was made in case we want to insert a new ophthalmologist to the ophthalmologist table in the database just by calling this procedure and giving it all the values of the parameters. |
|  | UpdateOphthalmologistDegree | This procedure takes two parameters which are the ophthalmologist id and the new degree (updated degree),  This procedure was made to update the medical degree of ophthalmologist in the ophthalmologist medical degree table when the inserted id matches the id which is in that table. |
|  | DeleteOphthalmologist | This procedure takes just one parameter which is the ophthalmologist id so we can know which ophthalmologist we want to delete based on the provided id parameter.  This procedure was made in case if we want to delete a ophthalmologist from the ophthalmologist table. |
|  | GetOphthalmologistDetails | This procedure takes two parameters which are the ophthalmologist username and password and returns details about the ophthalmologist.  This procedure was made if we want to see all the information that are related just for one ophthalmologist.  **Note:** each ophthalmologist has a username and a password so this function just returns the information of the ophthalmologist that his username and password matches the inserted username and password parameters. |

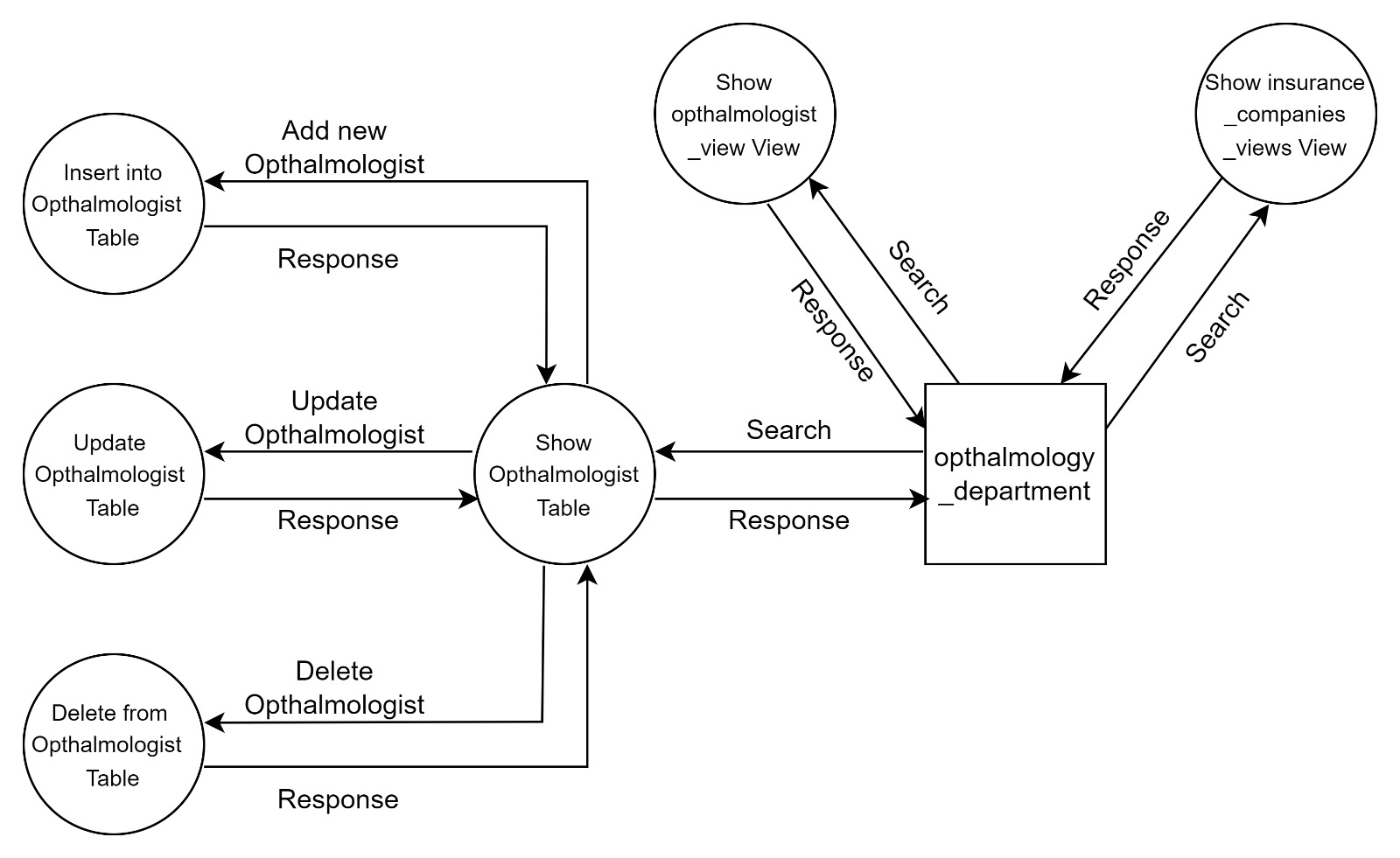
## Security

|  |  |  |  |
| --- | --- | --- | --- |
| **User name** | **Privilege Command** | **Description** | **Screenshot** |
| Ophthalmologist\_admin | GRANT SELECT ON  Ophthalmology\_Department.Ophthalmologist\_View TO 'Ophthalmologist\_admin'; | The Ophthalmologist admin user have a privilege to select and see all the data of the ophthalmologists that are in the ophthalmologist view table |  |
| GRANT SELECT ON  Ophthalmology\_Department.Insurance\_Companies\_view TO 'Ophthalmologist'; | The Ophthalmologist admin user have a privilege to select and see all the data of the insurance companies that are in the insurance companies view table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department. InsertOphthalmologist TO 'Ophthalmologist\_admin’; | The Ophthalmologist admin user have a privilege to insert a new ophthalmologist to the ophthalmologist table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.UpdateOphthalmologistDegree TO 'Ophthalmologist\_admin'; | The Ophthalmologist admin user have a privilege to update the ophthalmologist degree in case of any changes happens on their degree and make this update on the ophthalmologist table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.DeleteOphthalmologist TO 'Ophthalmologist\_admin’; | The Ophthalmologist admin user have a privilege to delete any ophthalmologist from the ophthalmologist table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.UpdatePatientMedicalHistory TO 'Ophthalmologist\_admin'; | The Ophthalmologist admin user have a privilege update the patient medical history after every appointment and this update will happen on the patient medical history table. |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.GetOphthalmologistDetails TO 'Ophthalmologist\_admin'; | The Ophthalmologist admin user have a privilege to see any of the ophthalmologist information based on the username and password |  |
| GRANT SELECT, INSERT, UPDATE, DELETE ON  ophthalmology\_department.opthalmologist TO 'Ophthalmologist\_admin’; | The Ophthalmologist admin user have a privilege to update and delete and insert anything to the ophthalmologist table |  |
| Patient\_admin | GRANT SELECT ON  Ophthalmology\_Department.Patient\_View TO 'Patient'; | The Patient admin user have a privilege to see all the data of the patients that are in the patient view table |  |
| GRANT SELECT ON  Ophthalmology\_Department.Insurance\_Companies\_view TO 'Patient'; | The Patient admin user have a privilege to see all the data of the insurance companies that are in the insurance companies view table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.InsertPatient TO 'Patient\_admin'; | The patient admin user have a privilege to insert a new patient to the patient table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.UpdatePatientMedicalHistory TO 'Patient\_admin'; | The Patient admin user have a privilege update the patient medical history after every appointment and this update will happen on the patient medical history table. |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.UpdatePatientEmergencyContact TO 'Patient\_admin'; | The Patient admin user have a privilege update the patient emergency contact and this update will happen on the emergency contact table. |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.DeletePatient TO 'Patient\_admin'; | The Patient admin user have a privilege to delete any patient from the patient table. |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.UpdateFollowUpAppointmentByPatientId TO 'Patient\_admin'; | The Patient admin user have a privilege to update the follow up appointments of patients, the follow up appointment will be updated in the patient table |  |
| GRANT EXECUTE ON PROCEDURE  ophthalmology\_department.GetPatientDetails TO 'Patient\_admin'; | The patient admin user have a privilege to see any of the patient information based on the username and password |  |
| GRANT SELECT, INSERT, UPDATE, DELETE ON  ophthalmology\_department.patient TO 'Patient\_admin’; | The patient admin user have a privilege to update and delete and insert anything to the patient table |  |

## User Interface

### Flowchart and Data Movement Diagrams





### Interfaces Development

|  |  |  |  |
| --- | --- | --- | --- |
| **Page ID** | **Title** | **Description** | **Screenshot** |
|  | Ophthalmologist | This page has all the ophthalmologist and their information such as first name, last name, phone number, specialty, username, password.  Also, this page enables the ophthalmologist to do some operations such as insert, update, delete, and select (view). |  |
|  | Ophthalmologist\_view | This page has all the ophthalmologist’s information which are from the ophthalmologist table such as first name, last name, phone number, specialty, username, password.  in addition to some other information that were joined from the work schedule table such as the days and the time that the ophthalmologist will be in the hospital  this page enables the ophthalmologist admin user just to select (view) all the ophthalmologist’s information. |  |
|  | Insurance companies view | This page has all the insurance companies’ information such as insurance name.  this page enables the ophthalmologist admin user just to select (view) all the insurance companies’ information. |  |
|  | Patient | This page has all the patient and their information such as first name, last name, phone number, date of birth, username, password.  Also, this page enables the patient to do some operations such as insert, update, delete, and select (view). |  |
|  | Patient\_view | This page has all the patient’s information which are from the patient table such as first name, last name, phone number, specialty, username, password.  in addition to some other information that were joined from the insurance table such as the insurance name, and other data were joined from the emergence contact table and other data were joined from patient medical history and the treatment table  this page enables the patient admin user just to select (view) all the patient’s information. |  |
|  | Insurance companies view | This page has all the insurance companies’ information such as insurance name.  this page enables the patient admin user just to select (view) all the insurance companies’ information. |  |

# Maintenance

## Database recovery & backups

**Database Recovery:**

In the event of a failure, database recovery involves returning the database to its original condition. The reliability and consistency of the data must be maintained through this procedure. Database management solutions frequently keep track of all transactions in a record along with the impact they have on the database. This recorded file can be used to restore the database in the event of a system failure or other issue by performing finished operations and undoing unfinished ones (errors). (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Database Backups:**

The data in a database can be recovered and restored using a database backup, which is a copy of the database's contents. Differential, incremental, and full backups are just a few of the backup techniques that are available. To reduce the danger of data loss, the best strategy is to maintain a regular backup routine. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Benefits:**

**Data Protection:** Backups and database recovery are the main strategies for preventing data loss. You can use a backup to restore data to a before the state in the case of a system crash or data corruption. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Business Continuity:** Any failure in business operations may result in financial losses. By reducing downtime in the case of a disaster backups and recovery technologies guarantee company continuity. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Compliance:** Regulations in several sectors require businesses to secure sensitive data, which means setting up a recovery and backup plan. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Testing and Development:** Without affecting the integrity of the live database, testing and development environments may be made using backups. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

**Reducing the effects of hardware failure:** At some point, hardware breakdown is expected. When this happens, regular backups make sure that the data can be restored to new hardware with the least amount of downtime. (Humeniuk, 2023) (Makenzie, 2023) (Veritas, 2022)

Makenzie, makenzie (2023) *Data Backup and Recovery explained*, *NinjaOne*. Available at: https://www.ninjaone.com/blog/data-backup-and-recovery-explained/#:~:text=Data%20backup%20is%20the%20practice,digital%20information%20for%20your%20business. (Accessed: 20 June 2023).

Humeniuk, F. (2023) *Top 7 advantages of data backup and Recovery*, *FluentPro Software*. Available at: https://fluentpro.com/blog/top-7-advantages-of-data-backup-and-recovery/ (Accessed: 20 June 2023).

Veritas, V. (2022) *Backup and recovery of data: The essential guide*, *Veritas*. Available at: https://www.veritas.com/information-center/data-backup-and-recovery (Accessed: 20 June 2023).

## Database maintenance in general

A series of tasks carried out to maintain a database working successfully and efficiently is referred to as database maintenance. These operations include a range of tasks that contribute to a database's stability, dependability, and best performance.

**Some of these tasks:**

**Data cleaning and validation:**

Databases might amass redundant, inaccurate, or out-of-date information over time. Data quality and accuracy are improved by routine data cleansing. Duplicate entries might be eliminated, inaccurate data could be updated, and outdated information could be corrected as part of this procedure. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Remove Old Backups:**

In fact, if you don't remove outdated backups, your SQL Server's hard disks will rapidly get full, which can lead to a variety of issues. The DBA's job is to make sure that extra backups are routinely deleted from a SQL Server. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Monitoring and adjusting performance:**

This includes regularly monitoring the database's performance, spotting any performance problems, and taking action to improve it. This might involve managing disk space, updating or building indexes, and optimizing SQL queries. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Security Management:**

Database security includes avoiding unauthorized usage and access to the database. This involves managing permissions for users, making sure that sensitive data is encrypted, and defending against dangers like SQL injection attacks. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Updates and patches for the system:**

In order to ensure the database management system's reliability and security and enable new features and upgrades, it is essential to routinely install updates and upgrades. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Database Optimization:**

Databases may be improved for increased effectiveness and performance. Reorganizing the database, arranging the data, or changing configuration settings may all be necessary. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

**Note:** also the backup and recovery are included in the tasks.

**Benefits:**

A database is protected against loss in an emergency by routine backups.

Maintaining databases regularly can assist reduce downtime and defend against harmful assaults.

It makes it simpler for IT staff to recover from disasters impacting their system by setting up archival backups of crucial data in case disaster recovery is required.

By deleting unnecessary or redundant data, it improves database efficiency. On some systems, this could result in a decrease in time. It is essential to make sure databases are operating at their best.

Over time, it enables a business to stay up with modern technological and security challenges. You can protect your data by maintaining your database. (TL Dev Tech, 2023) ( Maintenance Tasks, 2022) (OfficeTools, 2022) (McGehee, 2021)

TL Dev Tech, T.D.T. (2023) *What is database maintenance?*, *TL Dev Tech*. Available at: https://www.tldevtech.com/what-is-database-maintenance/ (Accessed: 20 June 2023).

Maintenance Tasks, M.T. (2022) *General Database Maintenance Tasks*. Available at: https://www.ibm.com/docs/en/imdm/11.6?topic=administrator-general-database-maintenance-tasks (Accessed: 20 June 2023).

OfficeTools, O. (2022) *Database maintenance explained*, *OfficeTools*. Available at: https://www.officetools.com/knowledgebase/database-maintenance-explained/ (Accessed: 20 June 2023).

McGehee, B. (2021) *Database maintenance plans in ssms: An overview*, *Simple Talk*. Available at: https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/database-maintenance-plans-in-ssms-an-overview/ (Accessed: 20 June 2023).

|  |  |  |
| --- | --- | --- |
| **Maintenance Task** | **Frequency** | **Purpose** |
| **Backup** | Daily/Weekly/Monthly | to regularly backup data to avoid losing it in the event of a disaster. |
| **Recovery testing** | Monthly | to ensure that the backup data can be successfully and accurately recovered. |
| **Performance monitoring** | Daily/Continuous | to evaluate and improve the database's performance and fix any problems. |
| **Data Validation & Cleanup** | Weekly/Monthly | To confirm data correctness, eliminate duplication, and fix inaccurate data. |
| **Security Updates and Checks** | As needed/Daily | To preserve the database against dangers such as illegal access. |
| **System Updates & Patches** | As released/needed | to apply necessary updates and patches in order to maintain the database's performance, security, and stability. |
| **Disaster Recovery Planning** | Annually | to plan approaches for dealing with the worst-case situations with the least amount of data loss. |
| **Capacity planning** | Monthly | To make sure that a database can manage growing capacity and to predict future data expansion. |

# Testing

## Data Validation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number** | **Type** | **Table name** | **Attribute name** | **Description** | **screenshot** |
|  | All cases of PK | Patient | Patient id | All the PK attributes must be:  Unique: we mean in our case that there can’t be two patient that have the same id, each id is related just for one patient | as we will see that I have Ahmad as a patient and his id is 1 and as we know that the patient id is unique  now we will see what would happen if we inserted new patient with the same id  as we can see that we inserted a new patient with the name Saif and the id is 1 it is the same as Ahmad’s id, the result is  it gives me an error because as I said that the patient id is unique so there won’t be two patient with the same id |
| Not Null: we mean in our case that you can insert a patient to the table without giving him/ her an id, so you must give the patient an id so you can be able to insert the patient | As we know that the primary key which is the patient id must have a value because it is single way to know about which patient we are talking from its id.  So, if we inserted a new patient and we leave the patient id empty without filling.  The result will be:  That error because of the primary key can be empty it must have a value |
|  | All cases of FK | Patient\_medical history | Patient id | As you know that the patient id is a foreign key in the patient medical history and as you see that the patient id foreign key is on delete cascade which means if we deleted on of the primary keys in the patient id it will also delete the same id that it is foreign key in other tables | The patient table before deleting the patient with the id 2  The patient\_medical history table before deleting it the patient with the id 2 (as you can see that the patient id in this table is a foreign key)    this query was used to delete the patient  this Is the patient table after deleting the patient that his id is 2  this is the patient\_medical history table after deleting the patient that his id is 2 |
| Patient id | As you know that the patient id is a foreign key in the patient medical history and as you see that the patient id foreign key is on update cascade which means if we update on of the primary keys in the patient id it will also update the same id that it is foreign key in other tables | The patient table before updating the patient with the id 3 to 10  The patient\_medical history table before updating the patient with the id 3 to id 10 (as you can see that the patient id in this table is a foreign key)  this query was used to update the patient    this Is the patient table after updating the patient that his id is 3 to id 10    this is the patient\_medical history table after updating the patient that his id is 3 to id 10 |
|  | Unique | Emergency contact | Patient id (FK) | Unique: we mean in our case that there can’t be two emergency contacts for the same patient | as we will see that I have Ahmad as a patient that have a emergency contact and his id is 1 and as we know that the patient id is unique which means each patient can have just one emergency contact that means we will not see two emergency contacts refers for the same patient (have the same patient id).  Patient Ahmad (id=1) have rami Nsour as his emergency contact  So, if we inserted another emergency contact for the patient Ahmad how’s id is 1 the result will be: |
|  | Default | Patient | Gender | In case we missed to insert the gender of the patient or of the Ophthalmologist during filling information, the gender will be automatically filled by Male | Adding a new patient without giving the patient a gender  As we can see we added new patient with the id 100 and his name is saif but without giving the gender and her is the results :    As we can see that the gender was filled automatically without giving it |
|  | Not null | Patient | First name | In case we want to insert a new patient to the patient table, and we didn’t fill the patient first name, the new patient will not be inserted because this attribute which is the first name must not be null which means can’t be empty | We can’t leave the patient first name empty.  Soo we will insert a new patient to the table but without giving him a first name, here is the results |
| Ophthalmologist | Phone number | In case we want to insert a new ophthalmologist to the ophthalmologist table, and we didn’t fill the ophthalmologist phone number, the new ophthalmologist will not be inserted because this attribute which is the phone number must not be null which means can’t be empty | We can’t leave the ophthalmologist phone number empty.  Soo we will insert a new ophthalmologist to the table but without giving him a phone number, here is the results |
|  | Check | Patient | Insurance degree | As we know that all the endurance degree are always greater than 0 so this will check the number if it is greater than zero or less than zero, in case it is greater than zero it will be accepted but in case it is less than zero it will not be accepted | I will insert a new patient with the value 0 for the insurance degree, the result will be: |
|  | ENUM | Patient | Gender | I used the Enum in case to ensure that no invalid gender to be enter to the gender attribute in the patient table, so the gender will just accept Male or Female | I will create a new patient and I will fill the gender with ‘others’ value, the result will be |

## Output Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Query Description** | **Screenshot (query + result)** | **Result validation** |
|  | In this query I want to see how much ophthalmologists I have in the ophthalmologist table | query:  results: five | After going back to the ophthalmologist table and checking the number of the ophthalmologist manually I found that the result of the query is true. |
|  | In this query I want to see how much ophthalmologist that they are specialist in retina | Query:  Results: just one | After going back to the ophthalmologist table and checking all the specialties of the ophthalmologist manually I found that the result of the query is true I just have one doctor who is specialist in retina |
|  | In this query I want to see how much patients that their insurance degree is greater than 2 | Query:  Results: non  A picture containing text, font, logo, white  Description automatically generated | After going back to the patient table and checking the insurance degree of each patients manually I found that the result of the query is true I don’t have any patient with that his insurance degree is greater than 2  A screenshot of a phone  Description automatically generated with low confidence |
|  | In this query I want to delete the patient with the id=1 | Query:  Results: | After going back to the patient table and checking it manually I did not found any patient with the id=1 so the result of the query is true |

## Security Validation

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **User Name** | **Description of privilege/no privilege** | **Screenshot (query + result)** |
|  | Patient\_admin | The Patient admin user have a privilege to see all the data of the patients that are in the patient view table | A picture containing text, screenshot, font, line  Description automatically generated |
| The Patient admin user don’t have a privilege to see all the data of the ophthalmologist that are in the ophthalmologist view table |  |
| The Patient admin user have a privilege to see all the data of the insurance companies that are in the insurance companies view table |  |
| The patient admin user has a privilege to call the insert patient procedure to insert a new patient to the patient table |  |
| The patient admin user does not have a privilege to call the insert ophthalmologist procedure to insert a new ophthalmologist to the ophthalmologist table |  |
| The Patient admin user have a privilege to call the update patient medical history procedure to update the patient medical history after every appointment and this update will happen on the patient medical history table. |  |
| The Patient admin user have a privilege to call the update patient emergency contact to update the patient emergency contact and this update will happen on the emergency contact table. |  |
| The Patient admin user have a privilege to call the delete patient procedure to delete any patient from the patient table. |  |
| The Patient admin user does not have a privilege to call the delete ophthalmologist procedure to delete any ophthalmologist from the ophthalmologist table. |  |
| The Patient admin user have a privilege to call the update follow up appointment procedure to update the follow up appointments of patients, the follow up appointment will be updated in the patient table |  |
| The patient admin user has a privilege to call the get patient details to see any of the patient information based on the username and password |  |
| The patient admin user have a privilege to update and delete and select anything in the patient table | Select:    Update:  Delete: |
|  | Ophthalmologist  \_admin | The Ophthalmologist admin user have a privilege to select and see all the data of the ophthalmologists that are in the ophthalmologist view table |  |
| The Ophthalmologist admin user does not have a privilege to select and see all the data of the patient that are in the patient view table |  |
| The Ophthalmologist admin user have a privilege to select and see all the data of the insurance companies that are in the insurance companies view table |  |
| The Ophthalmologist admin user have a privilege to call the insert ophthalmologist procedure to insert a new ophthalmologist to the ophthalmologist table |  |
| The Ophthalmologist admin user does not have a privilege to call the insert patient procedure to insert a new patient to the patient table |  |
| The Ophthalmologist admin user have a privilege to call the update ophthalmologist degree procedure to update the ophthalmologist degree in case of any changes happens on their degree and make this update on the ophthalmologist table |  |
| The Ophthalmologist admin user have a privilege to call the delete ophthalmologist procedure to delete any ophthalmologist from the ophthalmologist table |  |
| The Ophthalmologist admin user have a privilege to call the update patient medical history to update the patient medical history after every appointment and this update will happen on the patient medical history table. |  |
| The Ophthalmologist admin user have a privilege to call the get ophthalmologist details procedure to see any of the ophthalmologist information based on the username and password |  |
| The Ophthalmologist admin user have a privilege to select and update and delete and insert anything to the ophthalmologist table | Select:    insert:    delete:  update: |

## GUI Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Page name** | **Description** | **screenshot** |
|  | Ophthalmologist: | The ophthalmologist admin user can use this page to view, delete, insert, update any ophthalmologist information or data.  As we can see in the screenshot the red bottom is to insert new ophthalmologist, and the blue bottom is to view each ophthalmologist information separately , the green bottom is to edit of update any ophthalmologist information (data), and the purple bottom is to delete the ophthalmologist | Add new:    View:    Update: |
|  | Ophthalmologist\_view: | The ophthalmologist admin user can use this page just to view all the ophthalmologist’s information, but he is not able to do any operation on this page.  As we can see in the screenshot there is not any bottom on the left side such as delete or insert or update, this user can just view the information |  |
|  | Insurance companies view: | The ophthalmologist admin user can use this page just to view all the insurance companies’ information, but he is not able to do any operation on this page  As we can see in the screenshot there is not any bottom on the left side such as delete or insert or update, this user can just view the information |  |
|  | patient | The patient admin user can use this page to view, delete, insert, update any patient information or data.  As we can see in the screenshot the red bottom is to insert new patient, and the blue bottom is to view each patient information separately , the green bottom is to edit of update any patient information (data), and the purple bottom is to delete the patient | add new:  view:  update: |
|  | Patient\_view | The patient admin user can use this page just to view all the patient’s information, but he is not able to do any operation on this page.  As we can see in the screenshot there is not any bottom on the left side such as delete or insert or update, this user can just view the information |  |
|  | Insurance companies view: | The patient admin user can use this page just to view all the insurance companies’ information, but he is not able to do any operation on this page  As we can see in the screenshot there is not any bottom on the left side such as delete or insert or update, this user can just view the information |  |

## Assess whether meaningful data has been extracted

Within the provided database that I have created, it shows a real clear view of any database that would be in any hospital, because it covers all the Important information of the patients and the ophthalmologist that any hospital would looks for.

Moving on to some reasons that shows that everything in the data base is meaningful:

The first reason because all the tables which are in the database are related to each other’s by the primary keys and foreign key so we can get to any table easily by efficient operations for example just by scanning the treatment table we can know each treatment refers to which patient and also we can know how is the ophthalmologist that gives this treatment.

Also, we can see that the database is fully completed because it has a lot of tables that contains a big range of important data goes through patient information, medical history, emergency contact until we arrive to the ophthalmologist information, ophthalmologist work schedule, and treatments, insurance companies, this will enable us to know any information we need.

Also, the constraints that are used in all the attributes such as not null, default, check ensure the integrity and the quality of the data that it will be inserted which means that it will have meaning and we can understand it.

Starting with the first point:

All kinds of relations were used in this data base such as the one to one that we can found it between the patient table and the emergency table in logic each patient can have just one emergence contact as you can see in the actual inserted data that it is inserted in the emergency contact table, one to many relationships that we can found it between the patient and the treatment also between the ophthalmologist and the eye examination table, in logic that each patient can have many treatments also we can see this in the actual inserted data in the treatment table and in the eye examination table, many to many relationships that we can find it between the patient and the ophthalmologist table because relating to the logic that it is possible that one patient visits many doctors and one doctors see many patients and we can see this in the actual inserted data that we can find it in the Patient\_opthalmologist table (this table was made as a bridge between the patient table and the ophthalmologist table base it is a many to many relationship and also this table was made to ensure that it will be easy to delete or see or update any data across the related tables without losing the integrity of the data .

Everything in the database normalized as we can see that there is a table was made (patient\_ophthalmologist) between the patient and the ophthalmologist table because of the relationship which is many to many between the patient table and the ophthalmologist table which means that this new table was not came from space, it was must to made because of the many to many relationships that was found between these two tables.

Also, all the multi values attributes was taken into new tables so we don’t have duplicated values in the tables such as the medical history of patient attribute which means that each patient can have many medical histories, this attribute was taken into new table (patient\_medical history) also we can see that it is a multi-value table from the inserted data that the new table contains. Also there is another multi value which is the ophthalmologist medical degree so that why we made a new table ophthalmologist\_medical degree so we can put just the medical degrees in it because in logic each ophthalmologist might have many medical degrees and we can see this in the actual inserted data in this new table that shows the multi values (the purpose of making the Patient\_medical history and the Opthalmologist\_medical degree is because the multi values of these attributes that might be inserted).

Moving on to the views virtual tables that were made, these tables was very helpful for extracting meaning full data, these tables let us to define what the user can see form the real table, and also these kind of tables enables us to join more information from another tables so the user don’t need to move between more than table to get all the information that relates to the same patient or ophthalmologist. Also, these tables were made to make the queries easy to write rather than writing the query multiple times. And these tables also were made to raise the security by giving privileges to the user who can see information.

Moving on to the procedures to make the repeated operations that might happen every minute in the hospital more easer to do with a simple query rather than complex queries to save time and to save time donning the operations so these procedures will lead to finish the task easier. Also the procedures was used because it enables us to make joins between the tables such as getPatintInformation this procedure enable us to get all the data of a single patient from all the tables with a single simple query without taking a lot of time.

The views tables and procedures have a meaningful of the data that was extracted.

## Assess the effectiveness of testing

Testing was one of the main things in my database project when it was built, I have used two manual types of testing: the unit testing method and the end-to-end testing method.

Starting with the unit test method, I used this method on multi parts, the first part is to test all the primary keys which are in the table, the first test that was done on the primary key is to ensure that the primary key is unique which means there should not be any duplicated value in the column, so testing the uniqueness was so effective to ensure that the data base do not allow adding any duplicated values for the primary key. Also, I tested the primary key not null as a must that the primary key must have a value, so testing the not null was so effective to ensure that the data base do not allow I primary key with a null value. Also, I tested in case any of the primary keys were deleted all the record of this primary keys even if it is a foreign key in other table must be deleted. This is how the unit testing method was so effective to make sure that the primary key is working correctly without any mistake and hopefully all the primary keys were correct excepted one primary key I will mention it in the end of this question.

Also, I tested all the foreign keys to ensure that all the relations that are between the tables are correct, so the by using the unit testing method knew that all the tables are linked correctly. And I tested the check constraint to avoid any unexpected value that might be inserted, and I tested the default constraint, and the unit test was so effective so I can know if these constraints are working correct or not, by testing I got the expected outputs results which were correctly working.

Also, I tested all the given privileges that are given for both users: patient\_admin user and the ophthalmologist\_user, and I found that applying the unit test on this part was so effective because I make sure that each user has his own privileges by testing both scenarios, having this privilege and don have. Also, the unit test was so effective to check correctness of the privileges such as select, insert, and delete. And the unit test was so effective to check the security by checking the privileges limitation for both users which means each user can just make his tasks and the user can just see his tasks and can’t do anything to other users tasks, this is a necessary thing to avoid any issues. So, by using the unit test it was so effective to ensure that there is not any error in the privileges which means each user can make action on tasks that he need to do and can’t make any action an any task that he is not supposed to do, so because of the unit test I made sure that all the privileges are working correctly for each user. And the unit test was so effective to know that I have a high level of the security and to make sure that I don’t have any threats in my system.

Moving to the last unit test that I made, which was testing the users I tried to login multiple times on each user to check if there any issues and hopefully the unit test was very helpful to make me sure there are no issues with logging in using both users and everything was perfect.

Errors that I have notice because of the unit test that I have done, it was just one logical error which was between the patient table and the emergence contact table as I mentioned that the relation between these two table is one to one which means each patient can just have one emergency contact so will testing I found that the emergency contact table allows duplicated values which is not I want so I over checked and I found that I have to put the patient id attribute which is a foreign key in the emergency contact table as unique, so the unit test was so helpful and have a high effect on my database creation phase, this unit test method made me sure that every part I done while creating the data base is correct and also made me know what logical errors that I have in the spot before finishing everything.

Moving to the second manual type on testing, which it was the end-to-end testing method, it was so effective. I have logged in from the patient\_admin user and I tested all the privileges that are given to him such as select, delete, insert, and update and also I tested all the procedures that this user have a privileges on them, so using this type of testing allowed me to make sure that the login procedure does not have any issues and also that all the privileges that are given are working without anu issues. And I also did the same to the another user which is the ophthalmologist\_admin user and also I found that there is not any issues faced me or anu logical error.

# Evaluation of database solution

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

There was a lot of positive effect on the database solution that because of the user and system requirements.

Starting with how the user requirement effect the database solution, first thing by reading the user requirements I knew that we have two users the patient\_admin user and the ophthalmologist\_admin user so it was too easy to know how much users I need to create. Also the user requirement was so helpful to make me know what are the privileges that needed for each user, starting with the patient\_admin user I noticed based on the system requirements that the patient\_admin should be able to see their patients information such as their names and phone numbers, medical history and insurance details, treatment information, emergency contact information, and all these was achieved with a view table called patint\_view, and also based on the system requirement said that the patient\_admin user should be able to see all the insurance companies names that we have in this hospital, so an insurance company details view table was created to achieve this user requirement, so the user requirements was very helpful to know what view tables I should create.

Moving on to other part of the patient\_admin user requirement, I found based on the provided user requirement that the patient admin user should be able to insert new patients and delete patients and update the patient medical histories, also to update the patient follow up appointment, and update patient emergency contact, so the user requirement was very helpful to make me know what procedures I should make and what privileges I should give to this user, so based on the user requirement for the patient admin user I created a insert procedure that takes all the patient information, and I created a delete procedure that takes just the patient id, and I have created a update procedure for the medical history that takes the patient id and the new history, also I create an update procedure for the follow up appointment that takes the patient id and the new follow up appointment, and also I created a procedure to update the patient emergency contact that takes the patient id and the new contact information. All these procedures were made in a short time because of the positive effect of the provided user requirements. All the views tables and the procedures that were made was based on the provided user requirements and they effectively meet the requirements

Moving on to the second user which is the ophthalmologist\_admin user, it was very helpful to know what privileges should this user have based on the provided user requirements, so based on the user requirement I found that the ophthalmologist admin user should be able to see their ophthalmologist information such as the name, phone number, specialty, and the work schedule days, all these requirements was achieved by creating a view table called ophthalmologist view, also based on the system requirements said that the ophthalmologist admin user should be able to see all the insurance companies details that we have in the hospital, so another view table was created called insurance company details view table to achieve this user requirement, the user requirements was so helpful to make me create all the view tables in a short time without having any problem and they was straight toward and clear.

Moving in to the other part of the ophthalmologist admin user requirements, I found based on the provided user requirements that the ophthalmologist admin user should be able to insert, update, delete, view all the ophthalmologist information and also he should be able to update the patient medical history, so the user requirements was very helpful to make me know that procedures I should make and what privileges to give to this user, so based on the user requirements I created a insert ophthalmologist procedure that take the ophthalmologist id and all the other information, I have created a delete procedure that takes the ophthalmologist id, I have created a update medical degree procedure that takes the ophthalmologist id and the new ophthalmologist medical degree, also I have created a update patient medical history that takes the patient id and the new medical history of the patient and then I gave the ophthalmologist admin user a privilege to call all these procedures and view tables. All these procedures were made in a short time because of the positive effect of the provided user requirements. All the views tables and the procedures that were made was based on the provided user requirements and they effectively meet the requirements

## Suggested improvements



1-I would make a user’s table rather than having a username and password on both the patient table and the ophthalmologist table. So, I will create a user table that have the username and password as attribute and also, I will put the patient id and the ophthalmologist id as foreign keys so I can link the tables to gather.

2- I would add an email address attribute to the new user table.

3- I would create a phone number table instead of having a phone number in each table (patient, ophthalmologist). This new table will contain the phone number as an attribute and the patient id and ophthalmologist id as a foreign key.

4- I would create an address table so I can story the address of each patient in case of any emergence event happen.

5- I would create an appointment table so I can have a better track between the patient and the ophthalmologist. This table will contain the appointment id, patient id, ophthalmologist id, appointment date, appointment time.

6- Adding some attributes to the insurance table such as the coverage.

7- I would change the data type of the result of eye examination from varchar (200) to text.

## Evaluation based on improvements needed

Lets assume I did make a user table, this will decrease the redundant values rather of saving the username and password which are the login information in the patient table and the ophthalmologist table. also it will be easy to maintain because all the users login information will be in one table. This new table will also make everything easy in case we added a new human (user) to the database such as nurse this will be easy to add all the login information of the nurses just by adding the nurse id as a foreign key in the user’s table, which means it is easy expansion in case of any new role in the hospital.

Lets assume I added the email address attribute to the user table, this will help the patient and the ophthalmologist with communication and to reset their passwords in case they forget them.

Lets assume I did make a phone number table, this will decrease the redundant values instead of saving the phone numbers in both the patient table and the ophthalmologist table, also it will enable to save more the one number for each ophthalmologist in case he want to put the number of the landline and the number of the cell phone and the same thing for the patients in case they want to put mor that one number (home number, work number). And creating the phone number table will make the data base better organized because all the phone number information is stored on one place.

Lets assume I did make an address table, in this way the patient will be able to put his address and also he can put more than one and this table will help the hospital to find the location of the patient in emergency events

Lets assume I did make a appointment table, this will enable a better management of the proses of appointment between the patient and the ophthalmologist so we can know if the appointment is scheduled, completed, cancelled, or missed.

Lets assume I added the coverage attribute to the insurance table, this will give a clear view for the ophthalmologist when he deal with the patient and he know everything about his insurance so the ophthalmologist can put the patient in view of how much every think will cost based on the clear information which is in the insurance table.

Lets assume I changed the data type of the attribute ‘the results of eye examination’ which it is in the eye examination table from varchar (200) to text, this will make a big positive deference because in case of one of the ophthalmologists like to write his repot of the eye examination result in very detailed this will cause a problem because varchar (200) will not be enough so the ophthalmologist will have difficulties with writing the result.