Table of Contents

***Note: Please refer to dbscript.txt for SQL syntax for the tables and their records**

CIS 9340 Project Proposal

Agenda:

1. Overview of the business
2. Information and user-specific requirement or special interest
3. Objective for the database development

1. Overview of the business

CUNY Clinic is a campus clinic for Baruch students in New York City. It has doctors with different specialties, who treat patients with various illnesses and diseases. The clinic has been using a paper-based system where patients' records are kept in physical charts since its commence in 1970. With CUNY Clinic expanding rapidly in recent years, there is an increasing number of patients visiting the clinic and staff working here. In the meantime, the storage space required to maintain patients' records is running out. The management of CUNY Clinic is thus struggling with increasing complaints about the inefficiencies caused by the outdated system.

For instance, whenever doctors want to access their patients' records, they have to search through a collection of files containing patient information, past visits, and past treatments. This time-consuming process makes it difficult for doctors to locate key information, especially when the patients have complex medical conditions and chronic diseases. It is also difficult for the administrative staff to schedule visits between doctors and patients. There have been complaints about long wait times during visits, as well as insufficient facetime with doctors due to these administrative redundancies. Therefore, CUNY Clinic is transitioning to a new database management system (DBMS) for its stakeholders (discussed in *Information and user-specific requirement* section) to quickly and easily access the information they need, and for the clinic to solve the challenges it is currently facing.

2. Information and user-specific requirement

There are several requirements for this project:
(1) providing patients with the ability to view their medical records and doctor's notes,
(2) providing the doctors with the ability to view their patients' records, and to enter information regarding their patients,
(3) providing full access to the management only

Stakeholder	Requirements
Patients	● Access their own medical profiles which encompass o Patient's basic information such as SSN, DoB, phone, address, etc. o Insurance type o Patient's family medical history, if available o History of allergy o Doctors seen (Primary Care Provider / Specialists) o Appointments/scheduling ● Access the doctor's notes which entails o Symptoms o Diagnosis o Treatment (medication prescribed)
Doctors	● Doctors information: Doctors' ID, Available Schedule, Insurances Accepted ● Access their own patients' medical record (including medical history) (discussed above)

	● Doctors may input private notes (not available to patients)
	● Enter their own notes that patients have access to (discussed above)
	  o Enter diagnosis
	  o Enter prescription

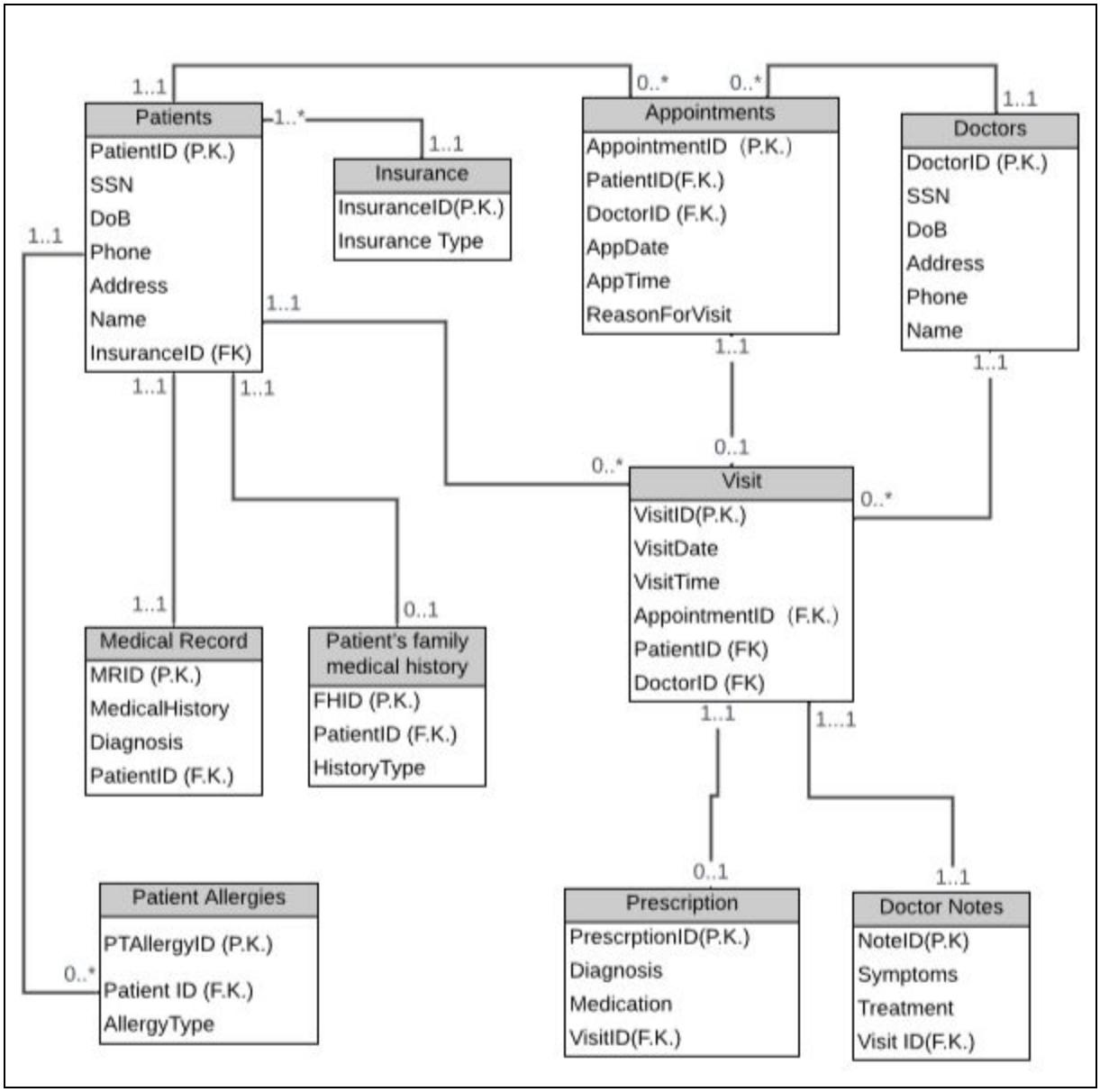
Management	● Access all patient records (de-identified for confidentiality) ● Access all doctors' notes (de-identified for confidentiality) ● Add doctors and patients to the database

3. Objective for the database development

The database is needed to help the team at CUNY Clinic, including managers and doctors, improve the efficiency of operations here. This goal can be reached by 1) streamlining the process of scheduling visits between patients and doctors, 2) making patient records easily accessible to the respective doctors, and 3) allowing for accurate updates of patient and doctor information. By streamlining the scheduling process, the clinic can minimize appointment mix-ups and increase patient satisfaction. When doctors can easily access their patients' files in the database, they will have more time for patients in each appointment, which in turn will improve quality of care. Additionally, maintaining accurate patient and treatment information is crucial in ensuring that each patient is receiving the intended treatment. For instance, if a patient is prescribed a certain medication, the database would maintain a record of what medication this patient is to receive each time the patient visits CUNY Clinic. If, for some reason, the patient wants to change the doctor who s/he sees and requests different medication prescribed, the database needs to be updated accordingly.

Having used a paper-based system for its patients, CUNY Clinic does not have the necessary tools to enter patients' information in an electronic database. As a result, the database system should be designed so as to make the process of uploading patient information as seamless as possible. The current system makes it more difficult to collect information from the patients and increase the potential for errors when entering patients' information provided on paper into the computer. For example, if patients had the opportunity to enter their information electronically, the data would be less prone to errors. The new database will be designed with this objective in mind, which would most likely affect other departments and operations in the clinic.

ERD_V1


Patients
PatientID (P.K.)
SSN
DoB
Phone
Address
Name
InsuranceID (FK)
Insurance
InsuranceID(P.K.)
Insurance Type
Appointments
AppointmentID (P.K.)
PatientID(F.K.)
DoctorID (F.K.)
AppDate
AppTime
ReasonForVisit
Doctors
DoctorID (P.K.)
SSN
DoB
Address
Phone
Name
Visit
VisitID(P.K.)
VisitDate
VisitTime
AppointmentID (F.K.)
PatientID (FK)
DoctorID (FK)
Medical Record
MRID (P.K.)
MedicalHistory
Diagnosis
PatientID (F.K.)
Patient's family medical history
FHID (P.K.)
PatientID (F.K.)
HistoryType
Patient Allergies
PTAllergyID (P.K.)
Patient ID (F.K.)
AllergyType
Prescription
PrescrptionID(P.K.)
Diagnosis
Medication
VisitID(F.K.)
Doctor Notes
NoteID(P.K)
Symptoms
Treatment
Visit ID(F.K.)
1..1
0..*
1..*
0..1
1...1

Relational Model_V1

Patients (PatientID, SSN, DoB, Phone, Address, Name, InsuranceID)

Primary key = PatientID Foreign key = InsuranceID references primary key of Insurance

Insurance (InsuranceID, Insurance Type, PatientID)

Primary key = InsuranceID Foreign key = PatientID references primary key of Patients

PatientFamilyMedicalHistory (FHID, PatientID, HistoryType)

Primary key = FHID Foreign key = PatientID references primary key of Patients

Doctor (DoctorID, SSN, DOB, Address, Phone, Name)

Primary key = DoctorID

MedicalRecord (MRID, MedicalHistory, Diagnosis, PatientID)

Primary key = MRID Foreign key = PatientID references primary key of Patients

PatientAllergies (PTAllergyID, PatientID, AllergyType)

Primary key = PTAllergyID Foreign key = PatientID references primary key of Patients

Appointments (AppointmentID, PatientID, DoctorID, AppDate, AppTime, ReasonForVisit)

Primary key = AppointmentID Foreign key = PatientID references primary key of Patients Foreign key = DoctorID references primary key of Doctors

Visit (VisitID, VisitDate, VisitTime, AppointmentID, PatientID, DoctorID)

Primary key = VisitID Foreign key = AppointmentID references primary key of Appointment Foreign key = DoctorID references primary key of Doctors Foreign key = PatientID references primary key of Patients

Prescription (PrescriptionID, Diagnosis, Medication, VisitID)

Primary key = PrescriptionID Foreign key = VisitID references primary key of Visit

DoctorsNote (NoteID, Symptoms, Treatment, VisitID)

Primary key = NoteID Foreign key = VisitID references

primary key of Visit

Check for 3NF criteria
Functional dependencies for:

- **Patients**
  - PatientID → SSN, DoB, Phone, Address, Name, InsuranceID
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **Insurance**
  - InsuranceID → Insurance Type, PatientID
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **PatientFamilyMedicalHistory**
  - FHID → PatientID, HistoryType
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **Doctors**
  - DoctorID → SSN, DOB, Address, Phone, Name
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **MedicalRecord**
  - MRID → MedicalHistory, Diagnosis, PatientID
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **PatientAllergies**
  - PTAllergyID → PatientID, AllergyType
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **Appointments**
  - AppointmentID → PatientID, DoctorID, AppDate, AppTime, ReasonForVisit
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- **Visit**
  - The relation has no repeating group or partial functional dependency, this relation is in 1NF and 2NF.
  - A. VisitID → VisitDate, VisitTime
  - B. .AppointmentID→ PatientID, DoctorID
  - B will create a transitive functional dependency, so the relation Visit is currently not in 3NF. To make it 3NF:

Visit (VisitID, VisitDate, VisitTime, AppointmentID)

Primary key = VisitID Foreign key = AppointmentID references primary key of Appointment.

In other words, we need to remove the relationships between Visit - Doctor and Visit - Patient. We have updated the ER model for the database to reflect this change.

- Prescription
  - PrescriptionID → Diagnosis, Medication, VisitID
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.
- DoctorNotes
  - NoteID → Symptoms, Treatment, VisitID
  - There are no repeating groups, no observable or derivable partial functional dependencies or transitive functional dependency; therefore this relation is in 3NF.

For each of the relation in the **updated** CUNY Clinic Database:

- There are no repeating groups (the intersection of every column and row contains an atomic value), so the relations are all in 1NF
- There are no observable or derivable partial functional dependencies, so all of the relations are in 2NF
- There are no observable or derivable transitive functional dependencies, so all of the relations are in 3NF

Updated Relational Model for the CUNY Clinic Database

Patients (PatientID, SSN, DoB, Phone, Address, Name, InsuranceID)

Primary key = PatientID Foreign key = InsuranceID references primary key of Insurance

Insurance (InsuranceID, Insurance Type, PatientID)

Primary key = InsuranceID Foreign key = PatientID references primary key of Patients

PatientFamilyMedicalHistory (FHID, PatientID, HistoryType)

Primary key = FHID Foreign key = PatientID references primary key of Patients

Doctor (DoctorID, SSN, DOB, Address, Phone, Name)

Primary key = DoctorID

MedicalRecord (MRID, MedicalHistory, Diagnosis, PatientID)

Primary key = MRID Foreign key = PatientID references primary key of Patients

PatientAllergies (PTAllergyID, PatientID, AllergyType)

Primary key = PTAllergyID Foreign key = PatientID references primary key of Patients

Appointments (AppointmentID, PatientID, DoctorID, AppDate, AppTime, ReasonForVisit)

Primary key = AppointmentID Foreign key = PatientID references primary key of Patients Foreign key = DoctorID references primary key of Doctors

Visit (VisitID, VisitDate, VisitTime, AppointmentID)

Primary key = VisitID Foreign key = AppointmentID references primary key of Appointments

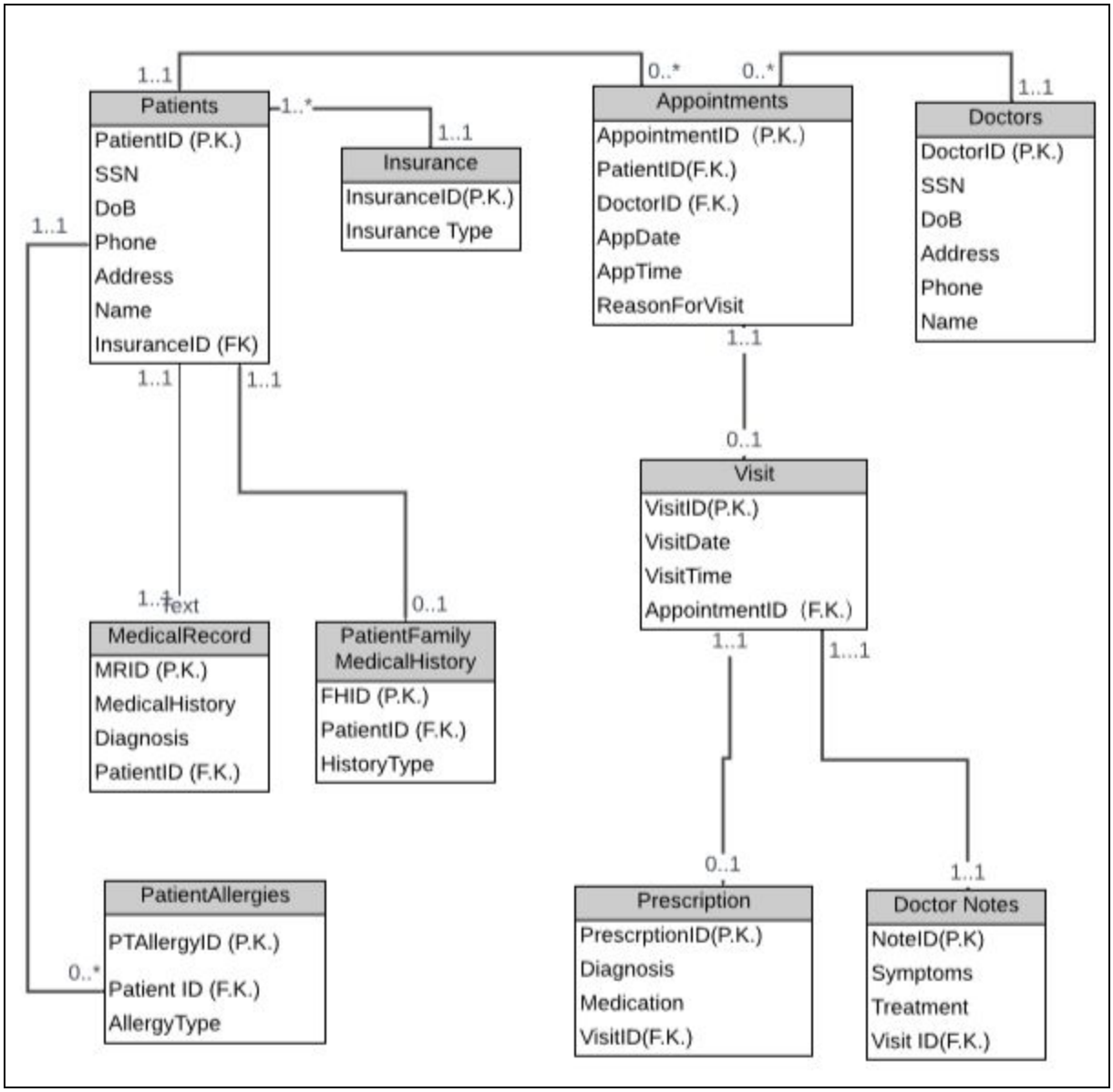
Prescription (PrescriptionID, Diagnosis, Medication, VisitID)

Primary key = PrescriptionID Foreign key = VisitID references primary key of Visit

DoctorsNote (NoteID, Symptoms, Treatment, VisitID)

Primary key = NoteID Foreign key = VisitID references primary key of Visit

ERD_V2, and the reasons for the change



Reason for change from ERD_V1 to ERD_V2:

The original table of Visit was not in 3NF. So we could either make this table to 3NF, or update the ERD. Below is the revised table of Visit (along with the process of making it in 3NF), which is now in 3NF. However, we decided to update the ERD to simplify the model.

The relation has no repeating groups or partial functional dependency, this relation is in 1NF and 2NF.

- A. VisitID → VisitDate, VisitTime
- B. AppointmentID → PatientID, DoctorID
- B will create a transitive functional dependency, so the relation Visit is currently not in 3NF.

To make it 3NF:

Visit (VisitID, VisitDate, VisitTime, AppointmentID)

Primary key = VisitID Foreign key = AppointmentID references primary key of Appointment

In other words, we need to remove the relationships between Visit - Doctor and Visit - Patient. We have updated the ER model for the database to reflect this change.

Relational Model_V2, and the reasons for the change

Reason for change: In Relational Model_V1, Patients table depends on Insurance table while Insurance table depends on Patients table; which was an oversight and these cannot be implemented. The updated relational model is shown below:

Insurance (InsuranceID, Insurance Type)
Primary key = InsuranceID

Patients (PatientID, SSN, DoB, Phone, Address, Name, InsuranceID)
Primary key = PatientID
Foreign key = InsuranceID references primary key of Insurance

PatientFamilyMedicalHistory (FHID, PatientID, HistoryType)
Primary key = FHID
Foreign key = PatientID references primary key of Patients

Doctor (DoctorID, SSN, DOB, Address, Phone, Name)
Primary key = DoctorID

MedicalRecord (MRID, MedicalHistory, Diagnosis, PatientID)
Primary key = MRID
Foreign key = PatientID references primary key of Patients

PatientAllergies (PTAllergyID, PatientID, AllergyType)
Primary key = PTAllergyID
Foreign key = PatientID references primary key of Patients

Appointments (AppointmentID, PatientID, DoctorID, AppDate, AppTime, ReasonForVisit)
Primary key = AppointmentID
Foreign key = PatientID references primary key of Patients
Foreign key = DoctorID references primary key of Doctors

Visit (VisitID, VisitDate, VisitTime, AppointmentID)
Primary key = VisitID
Foreign key = AppointmentID references primary key of Appointments

Prescription (PrescriptionID, Diagnosis, Medication, VisitID)
Primary key = PrescriptionID
Foreign key = VisitID references primary key of Visit

DoctorsNote (NoteID, Symptoms, Treatment, VisitID)
Primary key = NoteID
Foreign key = VisitID references primary key of Visit

Check for 3NF criteria

Note: Functional dependency is expressed with an arrow ‘’→” which means the determinant on the left functionally determines the dependents on th right

- **Insurance**
    - InsuranceID → Insurance Type
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **Patients**
    - PatientID → SSN, DoB, Phone, Address, Name, InsuranceID
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **PatientFamilyMedicalHistory**
    - FHID → PatientID, HistoryType
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **Doctors**
    - DoctorID → SSN, DOB, Address, Phone, Name
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **MedicalRecord**
    - MRID → MedicalHistory, Diagnosis, PatientID
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **PatientAllergies**
    - PTAllergyID → PatientID, AllergyType
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **Appointments**
    - AppointmentID → PatientID, DoctorID, AppDate, AppTime, ReasonForVisit
    - There is no repeating group; thus this relation is in 1NF.

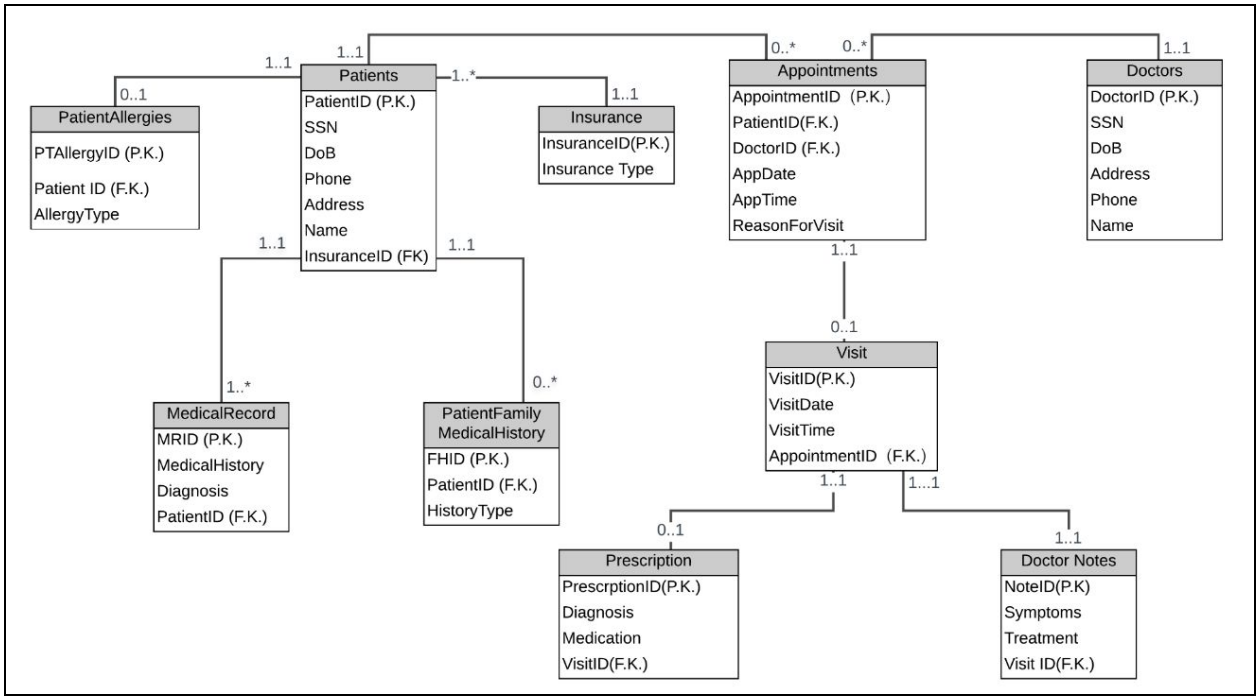
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **Visit**
    - VisitID → VisitDate, VisitTime, AppointmentID
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **Prescription**
    - PrescriptionID → Diagnosis, Medication, VisitID
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

- **DoctorNotes**
    - NoteID → Symptoms, Treatment, VisitID
    - There is no repeating group; thus this relation is in 1NF.
    - There is no observable or derivable partial functional dependency; thus this relation is in 2NF
    - There is no transitive functional dependency; thus this relation is in 3NF

ERD_V3, and the reasons for the change



Reason for change: Two pairs of one-to-one relationships were changed into one-to-many. The affected relationships are: Patients-MedicalRecord and Patients-PatientFamilyMedicalHistory. The changes were made to accommodate the data we populated into the database. For several patients in the database, there were more than one row of data for MedicalRecord and PatientFamilyMedicalHistory associated with that patient.

Queries, Reports, and their rationale/functionality

1. Patient with Allergy
Who are the patients with allergies, and which types of allergies do they have?

```
Select P.PatientID, PA.AllergyType
From Patients P, PatientAllergies PA
Where P.PatientID = PA.PatientID;
```

Patient with Allergy

PatientID	AllergyType
201	Food Allergy_Dairy
202	Food Allergy_Nuts
203	Food Allergy_Seafood
204	Drug Allergy_Steroids
205	Contact Dermatitis_Cosmetics

2. Number Of Patients With Diabetes In Family
How many patients have a family history of diabetes?

```
Select count(PF.PatientID) as NumberOfPatientsWithDiabetesInFamily
From PatientFamilyMedicalHistory as PF
Where PF.HistoryType = "Diabetes";
```

Number Of Patients With Diabetes In Family

NumberOfPatientsWithDiabetesInFamily
2

3. Most Popular Doctor
Who is the most popular doctor as measured by the number of appointments?

```
Select Name
From Doctor D,
(Select top 1 DoctorID, count(AppointmentID)
From Appointments
Group By DoctorID) T
Where D.DoctorID =T.DoctorID;
```

Most Popular Doctor

Name
ZOE JOY

4. Most popular Insurance
Which insurance company covers the most patients?

```
Select InsuranceType
From Insurance as I,  (Select Top 1 InsuranceID, Count(InsuranceID) From Patients Group by InsuranceID)
as T
```

Where T.InsuranceID=I.InsuranceID;

Most popular Insurance

InsuranceTy
Health First

5. High blood pressure & family medical history

List the Family Medical History for patients with high blood pressure.

Select HistoryType
From PatientFamilyMedicalHistory as F, MedicalRecord as M
Where M.MedicalHistory=”High Blood Pressure”
And F.PatientID=M.PatientID;

High blood pressure & Family Medical History

HistoryType
Heart Attack
Depression
Alzheimers

6. Same visit reasons and their old conditions

For people who came in for the same reasons, what were their old condition - diagnosis in their medical records?

SELECT ReasonForVisit, Diagnosis as Old_Condition
FROM MedicalRecord AS M, Appointments AS A, (SELECT A1.PatientID FROM Appointments AS A1, Appointments AS A2 WHERE A1.ReasonForVisit=A2.ReasonForVisit And A1.PatientID<A2.PatientID) AS Q
WHERE M.PatientID=Q.PatientID And A.PatientID=Q.PatientID;

Same visit reasons and their old conditions

ReasonForVisit	Old_Condition
Frequent urination	Hepatitis B
Frequent urination	Bladder Dysfunction
Frequent urination	Hepatitis B
Heartburn and abdominal pain	Stroke
Difficulty sleeping, lack of appetite	Stroke
Fever	Stroke
Heartburn and abdominal pain	Sports Injury
Difficulty sleeping, lack of appetite	Sports Injury
Fever	Sports Injury

7. Same visit reasons and their diagnosis and medication

Did people who came in for the same visit reason also receive the same diagnosis? List the diagnosis, medication, and reasons for visit for patients whose reasons for visit are the same. No need to list patient information.

```
Select DISTINCT P.Diagnosis, P.Medication, A1.ReasonForVisit
From Appointments A1, Appointments A2, Visit V, Prescription P
Where A1.ReasonForVisit = A2.ReasonForVisit
AND A1.PatientID < A2.PatientID
Order by A1.ReasonForVisit;
```

Same visit reasons and their diagnosis and medication

Diagnosis	Medication	ReasonForVisit
Anemia	B12 supplement	Fever
Common cold with sore throat	Ibuprofen 500mg	Fever
Deteriorating memory due to age	Ginko biloba	Fever
Indigestion	Ginger tea, probiotics	Fever
Light insomnia	Melatonin 3mcg	Fever
Pink eye	Eyedrops	Fever
Strep throat	Amoxicillin 2x/day for 5 days	Fever
UTI	Amoxicillin 2x/day for 5 days	Fever
UTI	Monurol 2x/day for 7 days	Fever
Anemia	B12 supplement	Frequent urination
Common cold with sore throat	Ibuprofen 500mg	Frequent urination
Deteriorating memory due to age	Ginko biloba	Frequent urination
Indigestion	Ginger tea, probiotics	Frequent urination
Light insomnia	Melatonin 3mcg	Frequent urination
Pink eye	Eyedrops	Frequent urination
Strep throat	Amoxicillin 2x/day for 5 days	Frequent urination
UTI	Amoxicillin 2x/day for 5 days	Frequent urination

8. Medication used in different diagnosis

Can the same medication be used in different diagnosis? If so, what medication and what diagnosis?

```
SELECT Distinct P1.Medication, P1.Diagnosis
FROM Prescription AS P1, Prescription AS P2
WHERE P1.Medication = P2.Medication
AND P1.Diagnosis <> P2.Diagnosis;
```

Medication used in different diagnosis

Medication	Diagnosis
Amoxicillin 2x/day for 5 days	Strep throat
Amoxicillin 2x/day for 5 days	UTI

9. No Shows

List patients who made appointments at CUNY Clinic but did not show up. What were their reasons for making an appointment?

```
Select p.PatientID, Name, Reasonforvisit
From patients p, appointments a, visit v
Where p.patientID = a.patientID AND
a.patientID NOT IN
(Select p.PatientID
From patients p, appointments a, visit v
Where p.patientID = a.patientID AND
```

a.appointmentID = v.appointmentID);

No Shows

PatientID	Name	Reasonforvisit	appDate
201	LEE KATE	Fever	5/1/2019
201	LEE KATE	Heartburn and abdominal pain	4/1/2019

10. Patient visits for same reasons

Who are the patients who make appointments for the same reasons?

```
SELECT DISTINCT P.Name, A1.PatientID,  A1.ReasonForVisit
FROM Appointments AS A1, Appointments AS A2, Patients AS P
WHERE A1.ReasonForVisit=A2.ReasonForVisit
And A2.PatientID<>A1.PatientID
AND A1.PatientID = P.PatientID
Order by A1.ReasonForVisit;
```

Patients who make appointments for same reasons

Name	PatientID	ReasonForVisit
ABBY LEE	210	Fever
LEE KATE	201	Fever
SUE LEE	205	Frequent urination
SUE MIKE	204	Frequent urination
SUE SAM	203	Frequent urination

Other Reports created from the tables directly, not from queries -

1. Patient List

Patient list report gives the management a clear summary of who the patients of CUNY Clinic has, along with their basic information.

Patients List

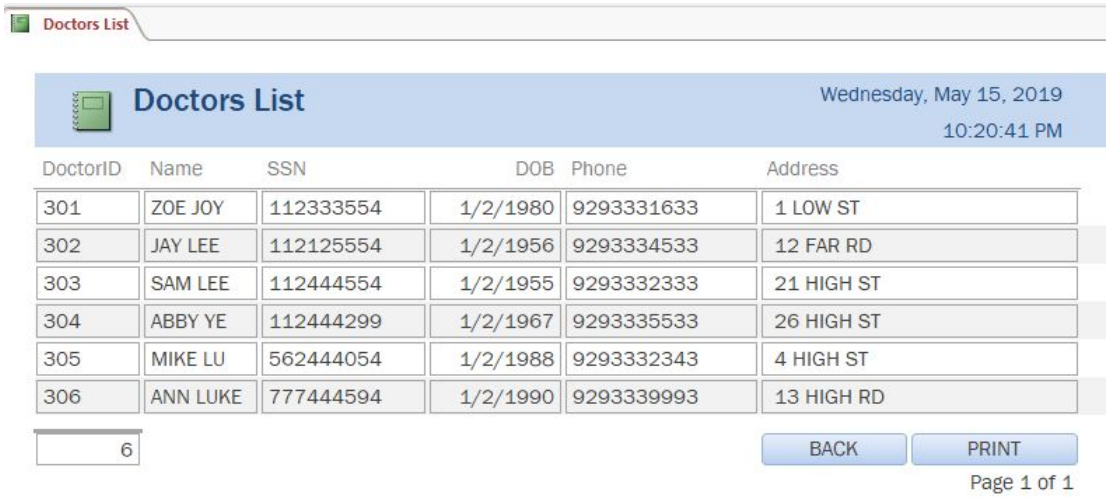
Patients List

InsuranceID	PatientI	Name	SSN	DoB	Phone	Address
101						
	201	LEE KATE	111222333	1/23/1990	9998887777	1 GREEN RD
	203	SUE SAM	111444555	1/23/1990	9293314445	23 WHITE RD
	206	LEE JOY	111777888	11/2/1990	9998887547	47 GREEN RD
	208	MAY ROSE	112222332	3/2/1960	9998882377	1 RED RD
	209	SKY YE	112333443	5/2/1970	9293335555	5 BLCK RD
102						
	204	SUE MIKE	111555666	11/2/1990	9293325007	4 WHITE RD
	205	SUE LEE	111666777	1/12/1995	9293318667	77 BLUE RD
	210	ABBY LEE	112444554	6/2/1950	9293331333	1 HIGH RD
	211	ZHAN	445556666	12/13/1999	3334345677	2 XX RD
	212	ZEN	111111111	5/9/2000	1212221212	2 XX RD
103						
	202	LEE MIKE	111333444	1/2/1990	9293337666	34 GREEN RD
	207	ROSE KATE	112000110	11/2/1995	9998887547	2 RED RD

Wednesday, May 15, 2019 Page 1 of 1

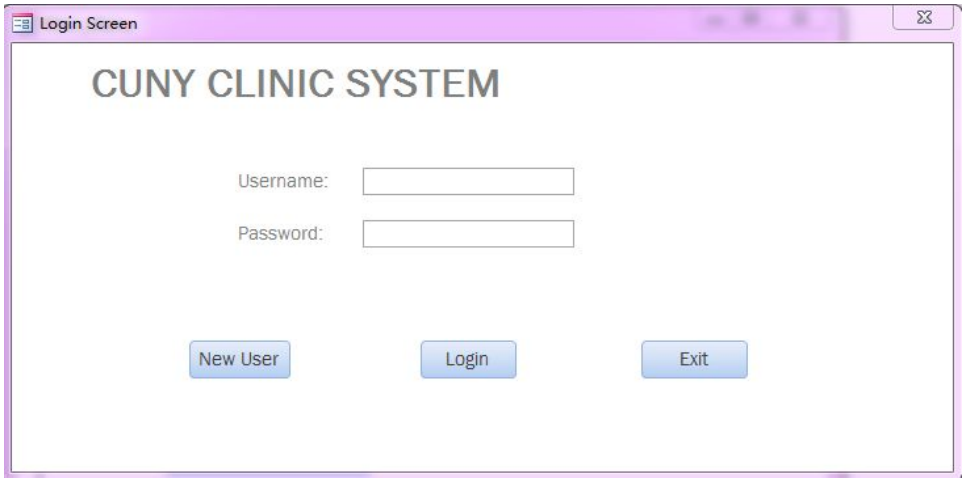
2. Doctors List

Similar idea to the Patient list report, Doctors List report gives the management an easy summary of CUNY Clinic's doctors.

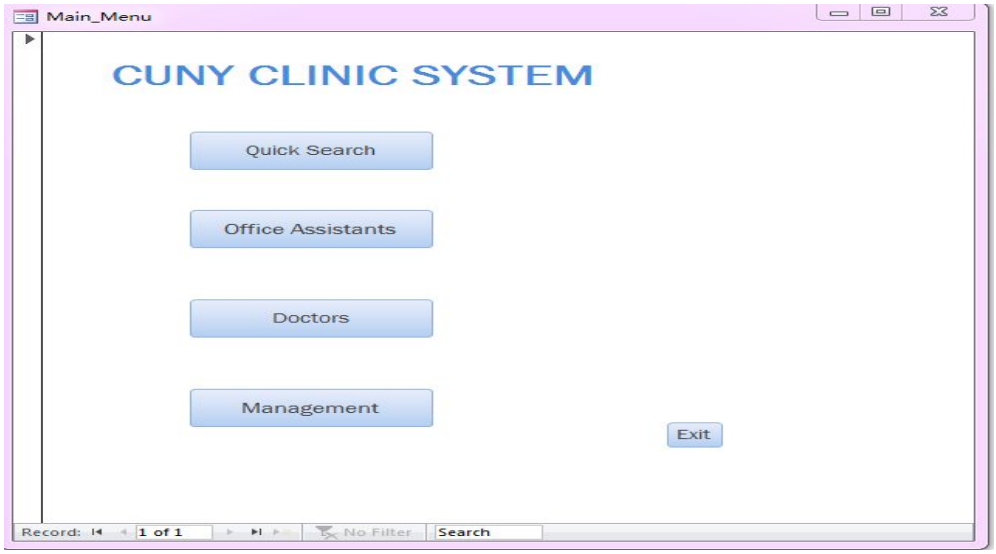


Applications and their rationale/functionality (including applications for forms and reports)

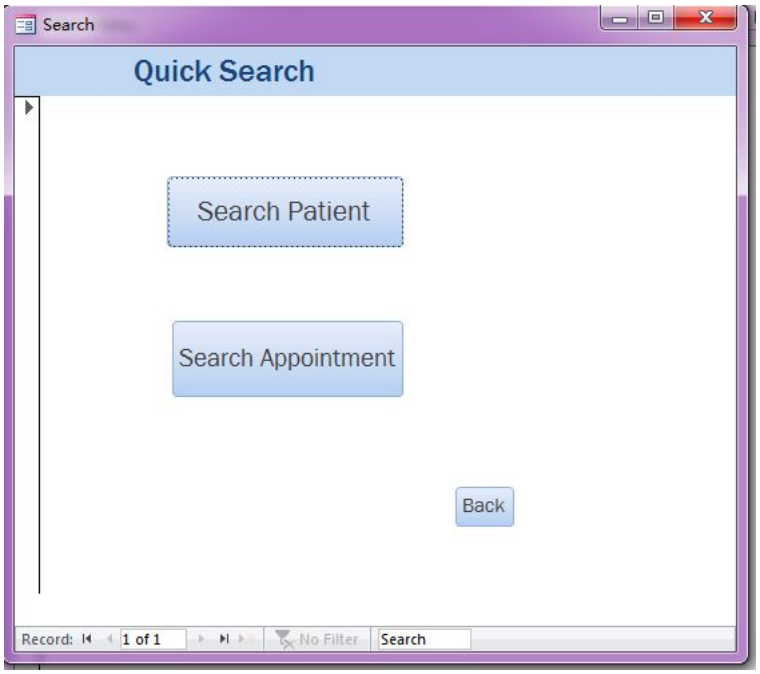
1. **In the application part, we design the "log-in" interface which corresponds to the "login list table" with vba behind the log-in button to enable it.**



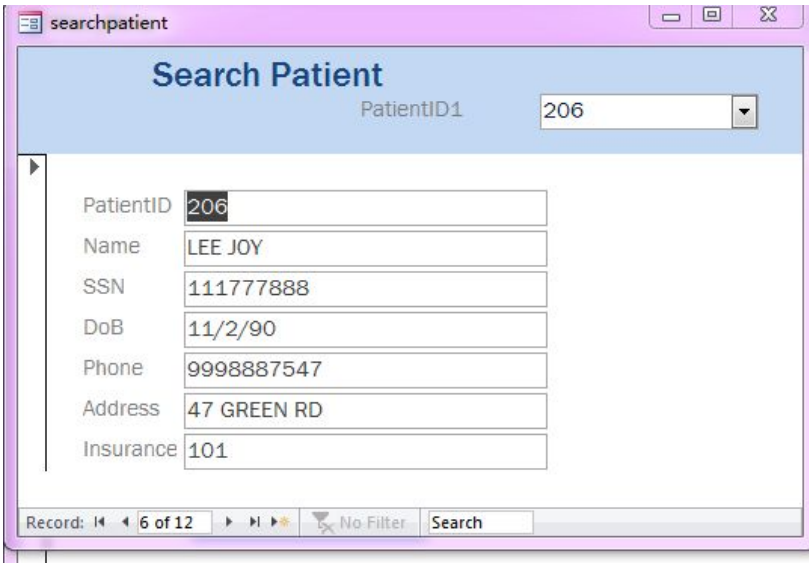
2. When users successfully log in to the system, they will be directed to the main menu, which is designed into the following four categories:



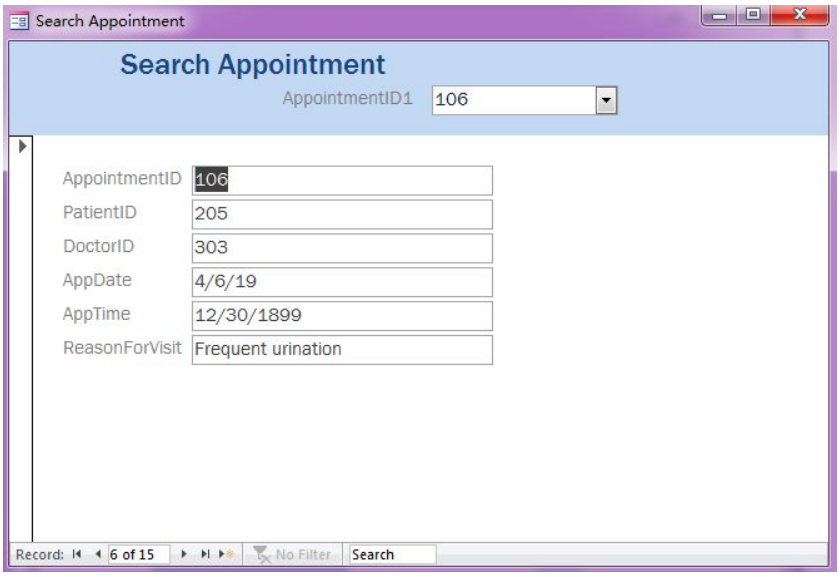
   a. **Quick Search: User can search the patients' information, appointment information quickly without referring back to the forms/tables**



- **Search Patient's information by typing/selecting the patient's ID**

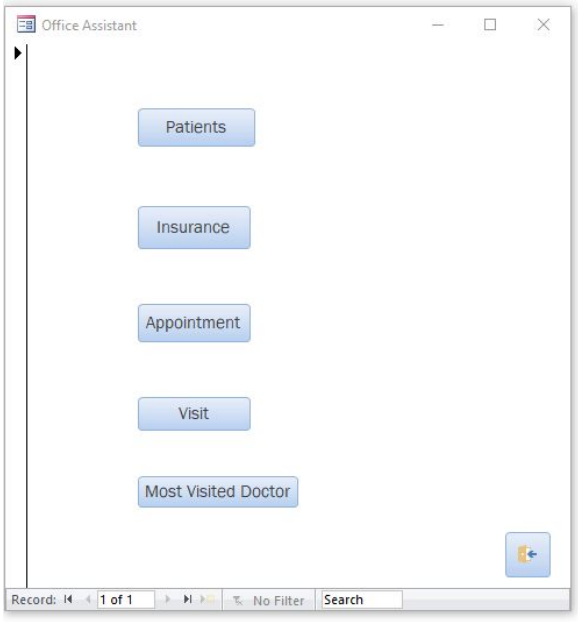


- **Search Appointment information by typing/selecting the patient's ID**

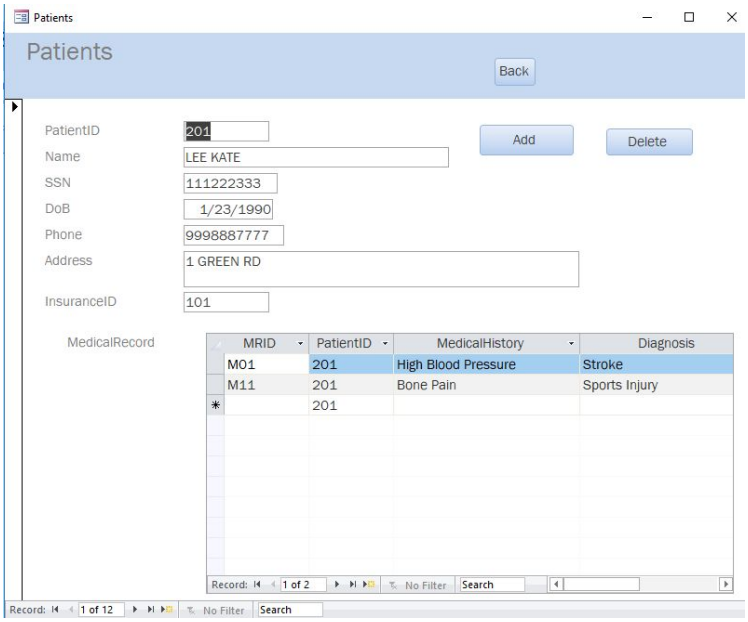


b. Office Assistant: Office assistants have access to the below forms/reports

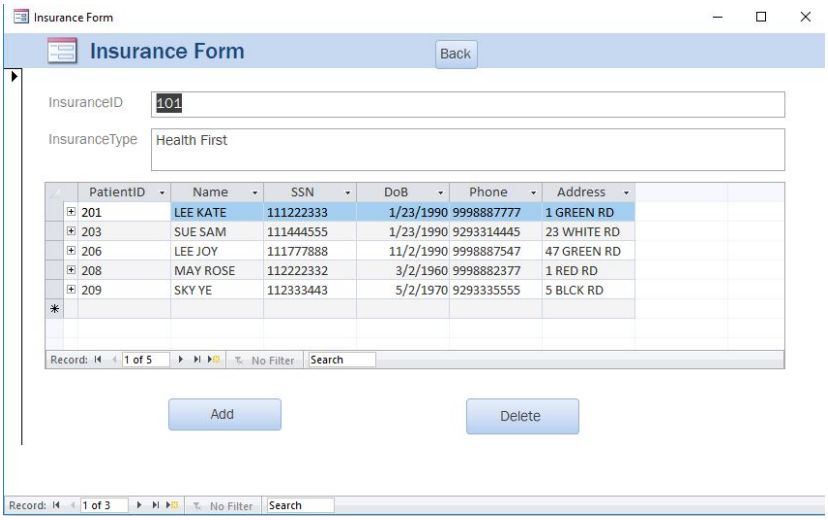
i. Office Assistant interface - Office assistants can access the below 4 forms for data edition and 1 report to report to the management



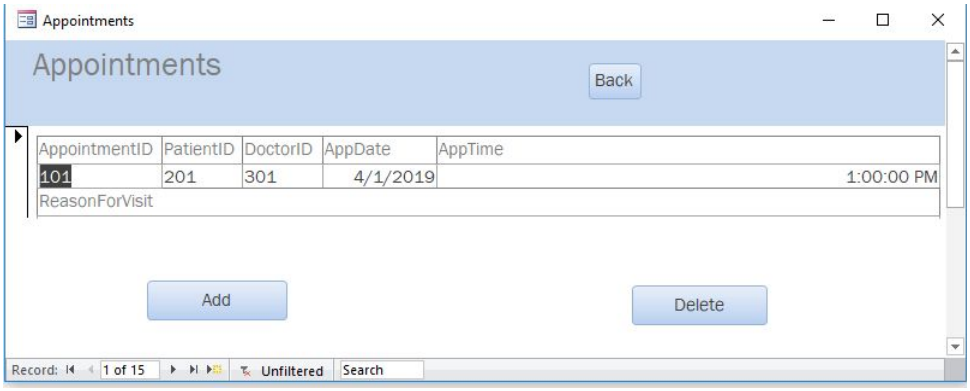
ii. Patients form - to add/edit/delete patient information



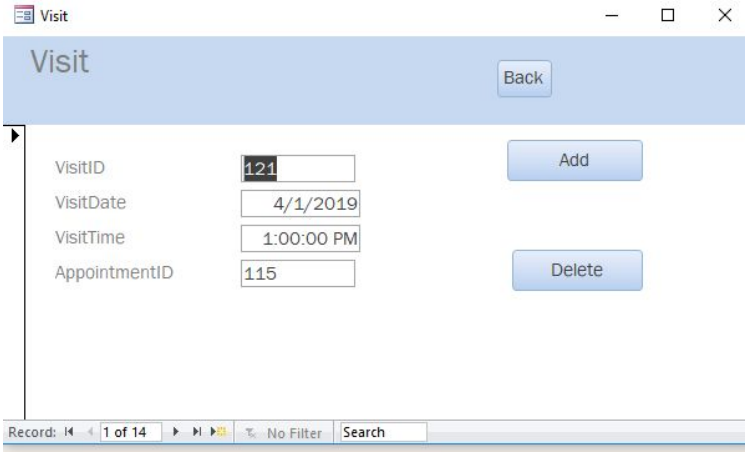
iii. Insurance form - to add/edit/delete insurance information



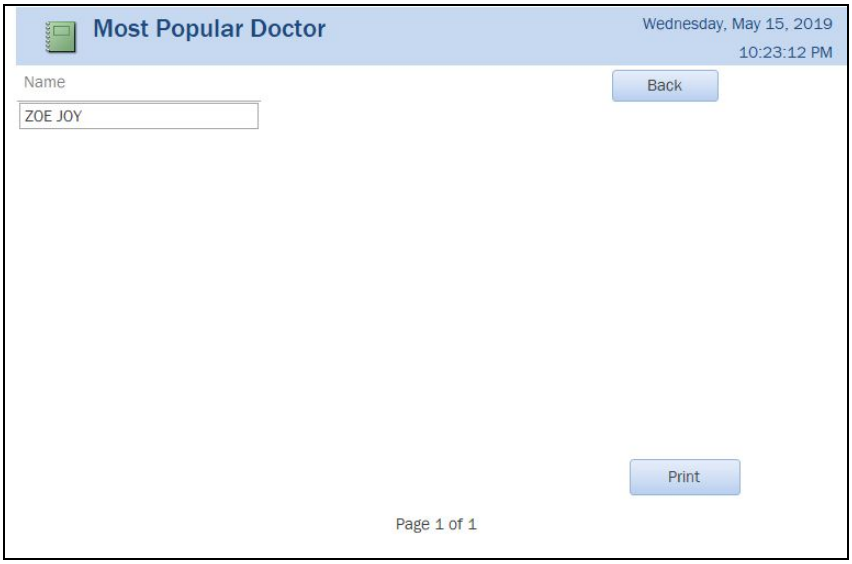
iv. Appointment form - to add/edit/delete appointments and their relevant details such as patientID, appointment date and time, and reason for visit



v. **Visit form - to add/edit/delete information associated with visits, which share the similar idea to the above mentioned Appointment form**

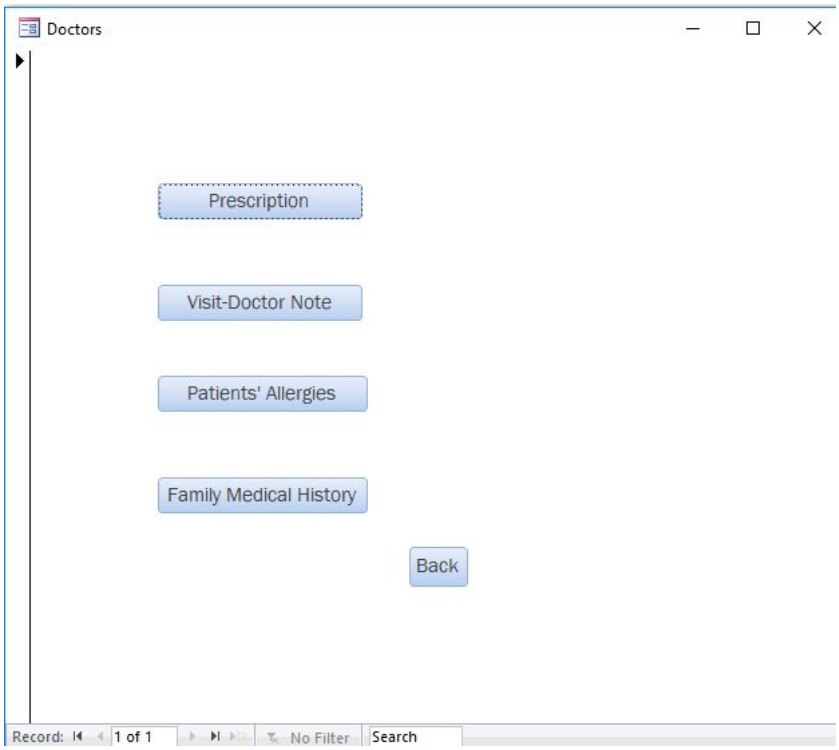


vi. **Most visited Doctor Report - to quickly know which doctor sees the most patient each day, and then report to the management**

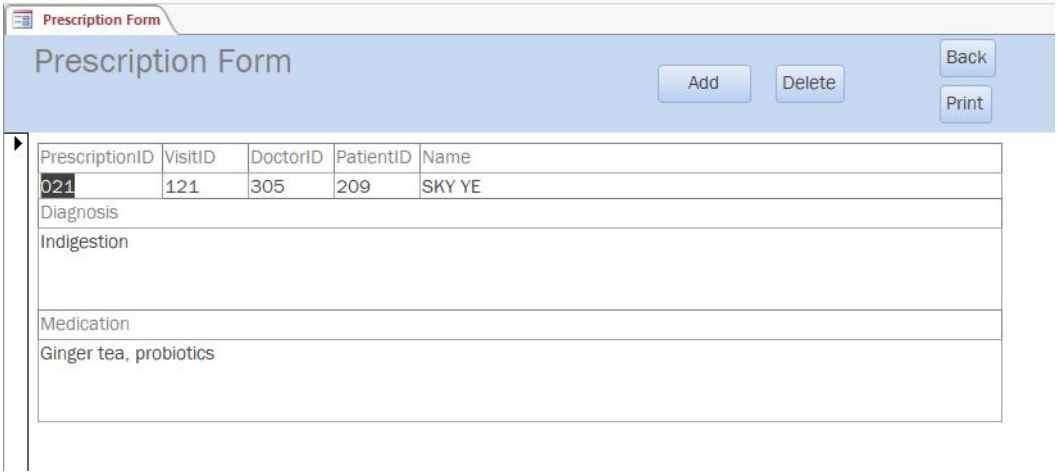


c. **Doctors: Doctors have access to the below forms**

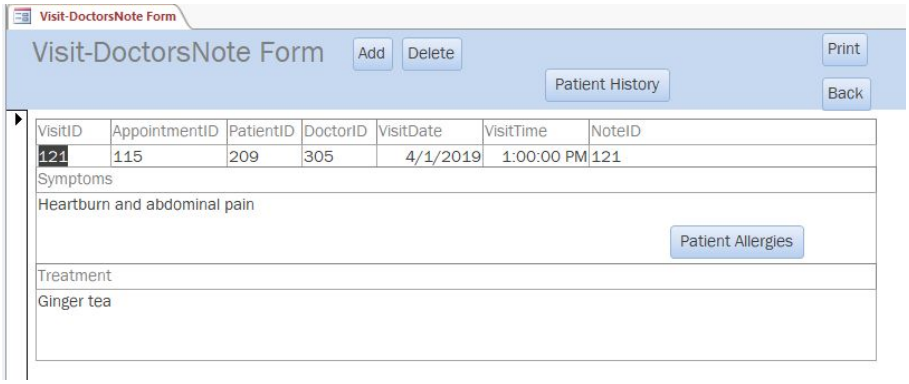
l. **Doctors interface - Doctors have access to the below forms to enter patient information when they see the patients, and to refer to patients' relevant information that would help the doctors with diagnosis and prescription**

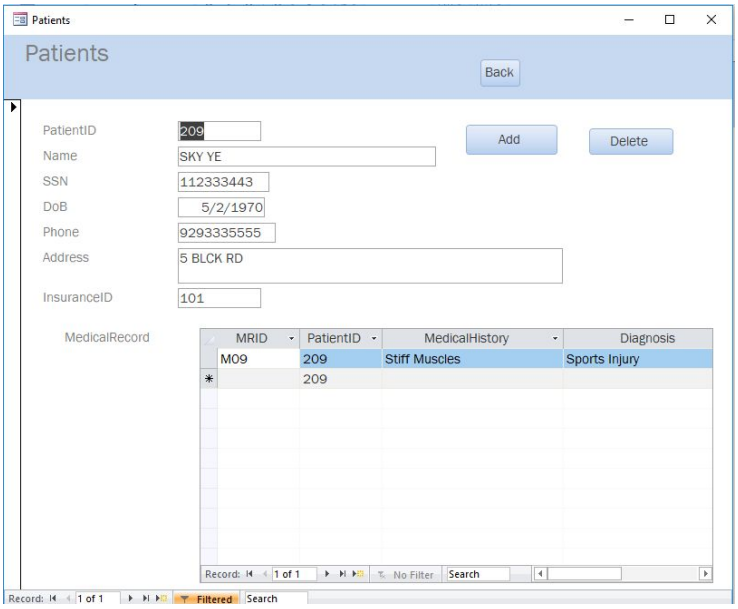


II. Prescription form - to add/edit/delete prescription information for patients

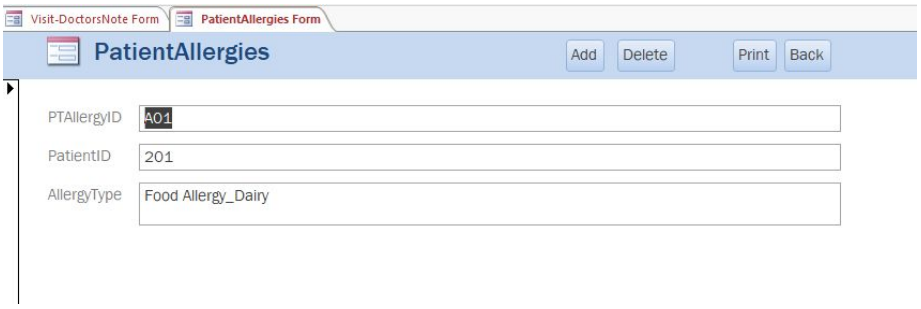


III. Visit-Doctor Note - to add/edit/delete doctors' notes used in see their patients. This form links to the forms of Patient History and Patient Allergies so doctors have quick reference to their patients' information in terms of past medical condition

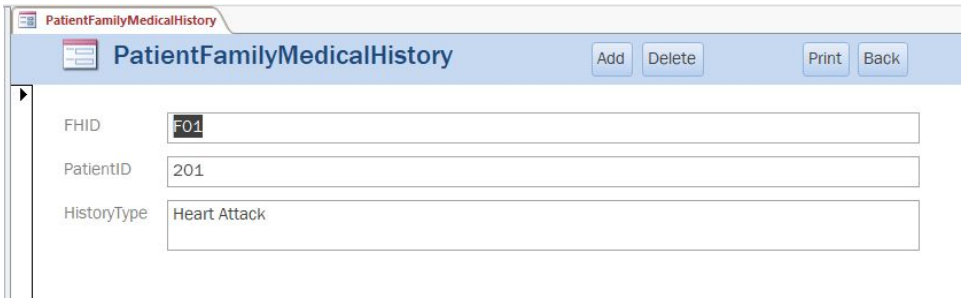




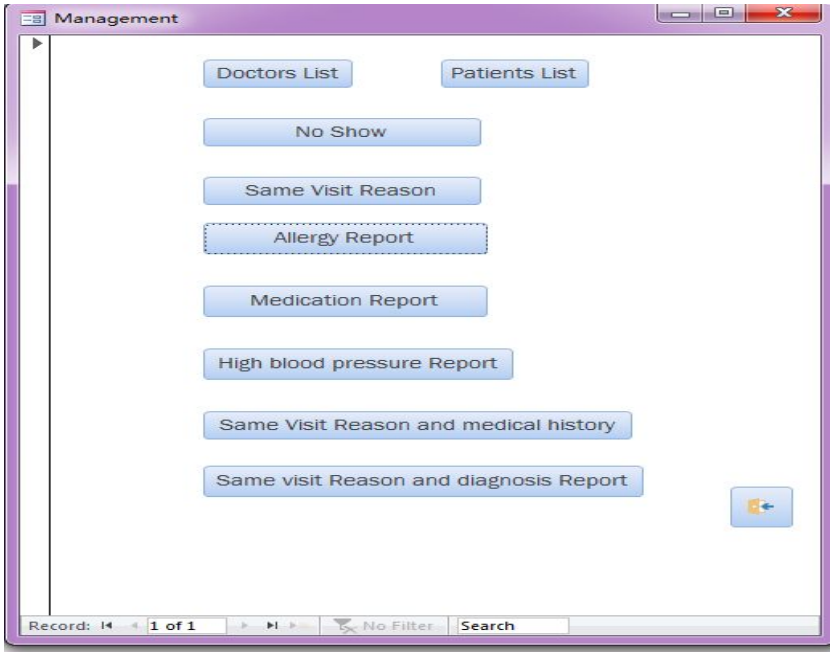
IV. Patients' Allergies form - to add/edit/delete patient's allergy information when applicable



V. Family Medical History form - to add/edit/delete patient's family medical history, if applicable



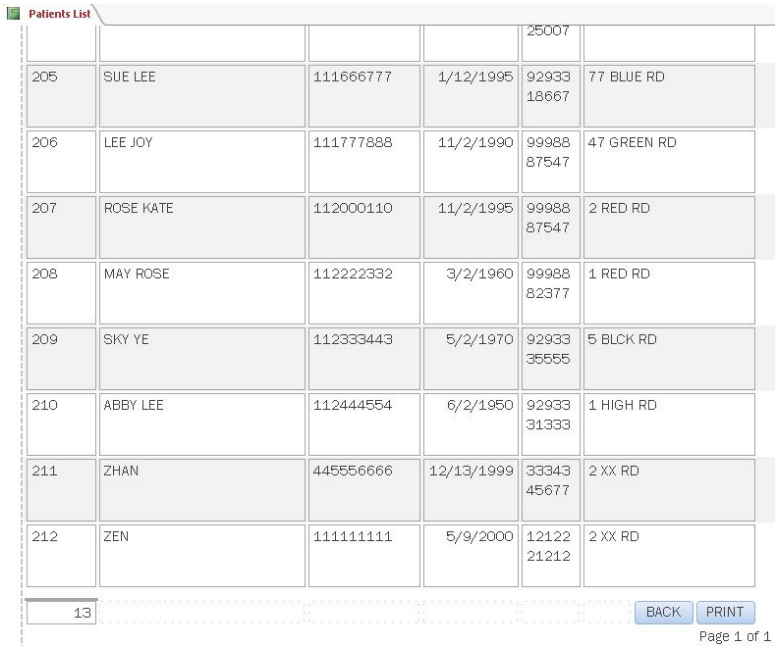
d. Management: Access to reports for clinic's daily operations



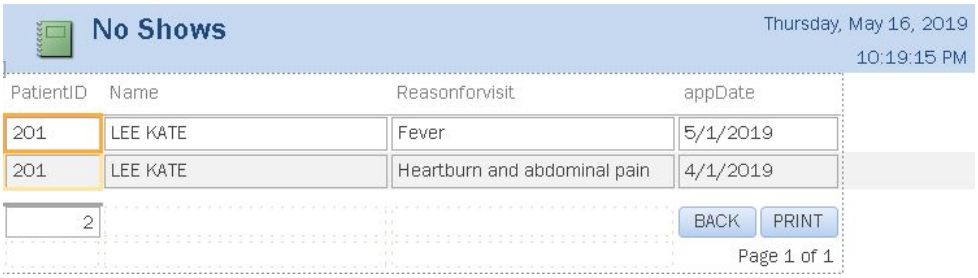
- **Doctors' List: Based on table - to display and print the list of current doctors the clinic has**

DoctorID	Name	SSN	DOB	Phone	Address
301	ZOE JOY	112333554	1/2/1980	9293331633	1 LOW ST
302	JAY LEE	112125554	1/2/1956	9293334533	12 FAR RD
303	SAM LEE	112444554	1/2/1955	9293332333	21 HIGH ST
304	ABBY YE	112444299	1/2/1967	9293335533	26 HIGH ST
305	MIKE LU	562444054	1/2/1988	9293332343	4 HIGH ST
306	ANN LUKE	777444594	1/2/1990	9293339993	13 HIGH RD

- **Patients' List: Based on table - to display and print the list of current patients the clinic has**

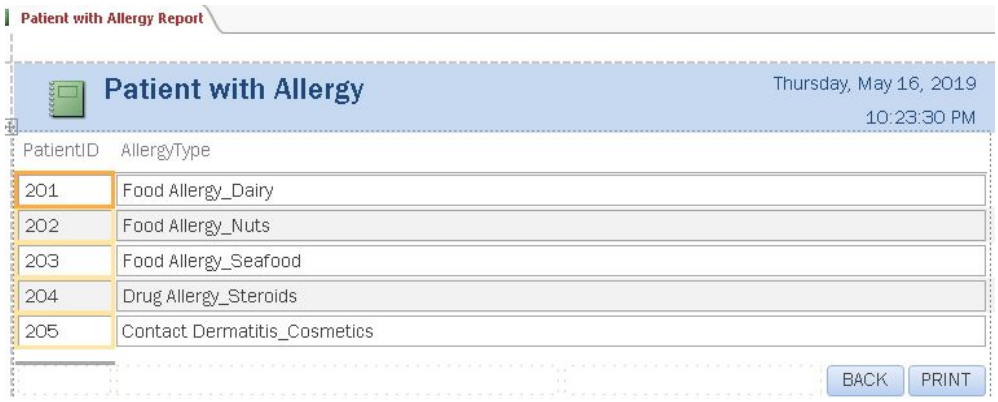


- **No Show: Based on Queries - to display and print the list of patients who did not show up**

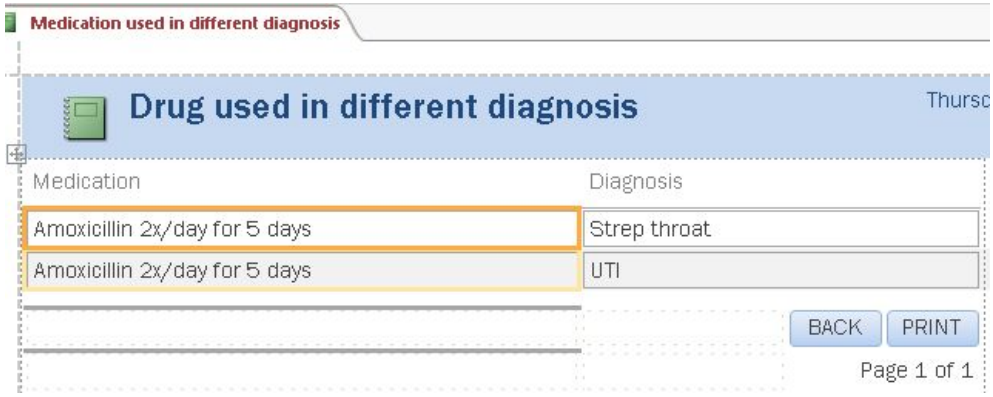


- **Same Visit: Based on Queries - to display and print the list of patients who make appointments for same reasons**

- **Allergy Report: Based on Queries - to display and print the list of who are the patients with allergies, and which types of allergies they have**

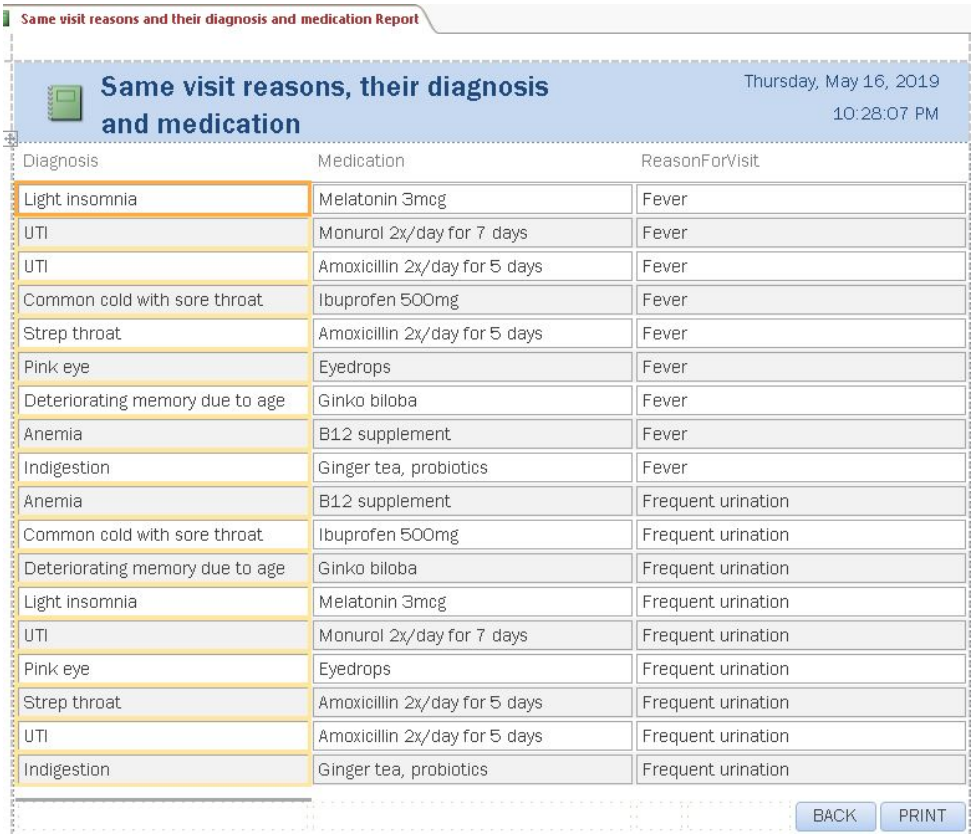


- **Medication Report: Based on Queries - to display and print the list of drugs used in different diagnosis**

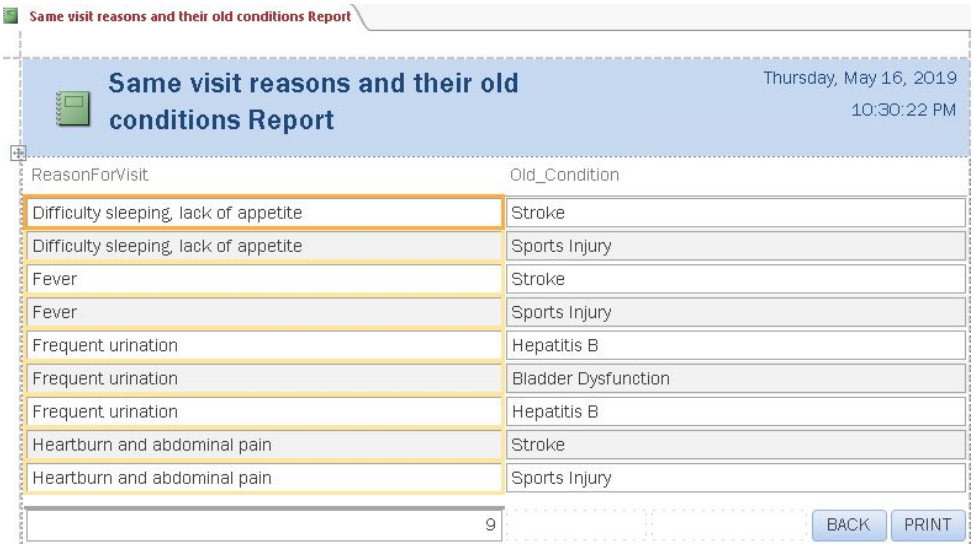


- **High Blood pressure Report: Based on Queries -to display and print the list of the Family Medical History (if available) for patients with high blood pressure**

- **Same Visit Reason and medical history report: Based on Queries - to display and print the list of the diagnosis, medication, and reasons for visit for patients whose reasons for visit are the same**



- **Same Visit Reason and diagnosis report: Based on Queries - to display and print the list of people who came in for the same reasons and their old condition (diagnosis in their medical records)**



Closing Remarks

Experience with the project

This project allowed us to leverage our knowledge of database management systems effectively. Having a clear project timeline and knowing which deliverables were required in the beginning, we were able to visualize the project outcome and adjust the scope of the project to accommodate the given timeline. Working on a real-world scenario business operation, in this case, a campus medical clinic, we had the opportunity to look at the business from every possible angle, from the perspective of a patient making appointments to that of a doctor writing prescriptions, or a manager who oversees the clinic. Since the project involves both designing and implementing a database system, we were able to put most of what we learned in the course into practice. Most importantly, we learned how to achieve all of the above while working as a team.

The most challenging aspect of the project

We found the implementation of the user application to be the most challenging, both due to our limited experience with Microsoft Access and the fact that building an application involves many interconnected layers that as a whole are prone to errors. During the implementation, we found that it was critical to design the most fundamental parts of the database correctly before moving on to more complicated steps. For example, while creating the Patients form, we had difficulty creating new records using the form, which was due to an error in our ER diagram. After we realized that the relationship between the Patient Entity and the MedicalRecord entity was one-to-many instead of one-to-one, we were able to choose the appropriate form design.

Another difficulty we had during the application building process was version control. Microsoft Access does not support collaboration between different computers, which made it extremely complicated for group members to work on the database at the same time. Even though we delegated different aspects of the application among group members, we still had to take turns editing the master database in order to integrate each member's work. This made the process unnecessarily time-consuming and ineffective.

The least challenging tasks

We found the reports in Access to be fairly simple to create. The graphic user interface in Access proved to be helpful in this particular task.

What we would have done differently

Due to time constraints, we decided to limit the scope of the database to the day-to-day appointments and treatment at the clinic. As a result, we did not include other operational aspects, such as insurance claims and billing. However, if time permitted, we would like to expand the scope to reflect the operations of a real clinic.