**MIS 6326 DATABASE MANAGEMENT**

Presented to Dr. Young Ryu

**LIFELINE HOSPITAL**

**RELATIONAL DATABASE MANAGEMENT SYSTEM**



**Present**

BY

SAI SUSHMA PALLE

POOJA BHAMARE

GOWTHAM PHANI CHANDRA KONDA

RAJESH KIMIDI

JAGADESH VARMA NADIMPALLI

**TABLE OF CONTENTS**

**1. PROBLEM DESCRIPTION…………………………………………………..1**

**2. ORGANIZATION INTRODUCTION………………………………………2**

**3. SCOPE OF DATABASE……………………………………………………….2**

* Patient Table………………………………………………………………………………2
* Physician Table…………………………………………………………………………..3
* Visit Table…………………………………………………………………………………..3
* Treatment Table…………………………………………………………………………3
* Appointment Table…………………………………………………………………….4
* Payment Table……………………………………………………………………………4
* Invoice Table………………………………………………………………………………4
* Insurance Table………………………………………………………………………….4

**4.ENTITY RELATIONSHIP DIAGRAM………………………………………5**

**5. RELATIONAL DATABASE SCHEMA…………………………………….6**

* Patient Table………………………………………………………………………………6
* Physician Table…………………………………………………………………………..6
* Visit Table…………………………………………………………………………………..6
* Treatment Table…………………………………………………………………………6
* Appointment Table…………………………………………………………………….7
* Payment Table……………………………………………………………………………7
* Invoice Table………………………………………………………………………………7
* Insurance Table………………………………………………………………………….8

**6. MENU SCREEN FORM……………………………………………………….9**

**7. DATA INPUT SCREEN FORMS…………………………………………….12**

**8. SAMPLE REPORTS……………………………………………………………..20**

**1.PROBLEM DESCRIPTION** :

* As the current system in Sai Memorial Hospital uses manual approach to store the information of patients, it is very likely to get misplaced and lost which could be problematic if in case the family members of the patients are to be informed about something important.
* Proper data storage is required for the treatment undergone by a specific patient, which can help to analyze the duration taken to recover by the patient and the medicines used for the upcoming intern doctors.
* Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost.
* A significant part of the operation of any hospital involves that acquisition, management and timely retrieval of great volumes of information.
* Most of the insurance companies use online approach for claims and funds transfer. So the patient has immediate emergency then the insurance company would need all the data to be received promptly and in a proper fashion. These manual entries would take time and could cost for someone’s life.
* All this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized Sai Hospital Database Management System will automate the management of the hospital making it more efficient and error free.
* All the appointments can be handled online which will make it easy to check if that physician is available for that time on that particular day.

**2.ORGANIZATION INTRODUCTION** :

The project is based on the information regarding the day to day activities conducted in the Sai Memorial Hospital. The hospital currently uses manual techniques to note down all the critical and important information regarding the patients, the doctors, the outstanding payments and the payments received successfully. The current system requires a lot of manual work with numerous papers work which is difficult to maintain at times.

Many times, the historic data is required when some information is to be retrieved. The information should be stored in such a way that it should be accessible easily and is not lost. The maintenance should be easy and flexible so that if any patient’s or doctor’s information has to be retrieved or maintained it should be done with ease.

As the hospital is a smaller set of units, they were trying to store all the relevant information manually. But as the days are passing, the many patients are getting admitted for some particular treatments and the manually way of maintaining the records is getting difficult. As the hospital is expanding now, the manual approach can be updated to online way which will be of ease for proper maintenance.

**3. SCOPE OF DATABASE :**

**1.PATIENT TABLE:**

This table consists of all the basic information of the patients. The information consists of their name, contact number, age and gender. The column which uniquely identifies every single patient is the patient #ID. It’s assigned for every new patient.

* PatientID **(Primary Key)**
* PatientFName
* PatientLName
* ContactNumber
* Age
* Gender

**2.PHYSICIAN TABLE:**

This table consists of all the information regarding the physician. It includes their name, contact number and the specialty they have. The column which uniquely identifies every physician uniquely is the physician #ID.

* PhysicianID **(Primary key)**
* Speciality
* PhysicianFName
* PhysicianLName
* ContactNumber
* DailyAvailableFrom
* DailyAvailableTill

**3.VISIT TABLE:**

This table consists of information of a visit made to the Sai Memorial Hospital. Every visit is associated with a unique Visit #ID.

* VisitID **(Primary key)**
* AppointmentID(**Foreign Key**)
* VisitDate

**4.TREATMENT TABLE:**

This table has all the information regarding what the results of the treatments were performed on the patients, including the date and time the treatments were carried on. The Treatment table has patient #ID, physician #ID and visit #ID to track every single treatment. Also, there is a unique treatment #ID to uniquely identify every treatment.

* TreatmentID **(Primary key)**
* PatientID **(Foreign key)**
* PhysicianID**(Foreign key)**
* VisitID **(Foreign key)**
* Results
* TreatmentDate
* TreatmentTime
* TreatmentType

**5.APPOINTMENT TABLE:**

This table has all the information regarding the appointment time and date. The appointment table also has patient #ID and visit #ID which is helpful to track the patient and the visit number associated with it. Every appointment can be uniquely tracked by the appointment #ID.

* AppointmentID **(Primary Key)**
* PatientID **(Foreign Key)**
* AppointmentDate
* AppointmentTime

**6.PAYMENT TABLE:**

This table keeps a track of all the payment received and type of payment by which the payment was received. It has an Invoice #ID linked to that specific payment and a unique payment #ID to identify every unique payment.

* PaymentID**(Primary Key)**
* InvoiceID**(Foreign Key)**
* CardType
* CardNo
* CardExpDate
* PaymentTime
* PaymentAmount

**7.INVOICE TABLE:**

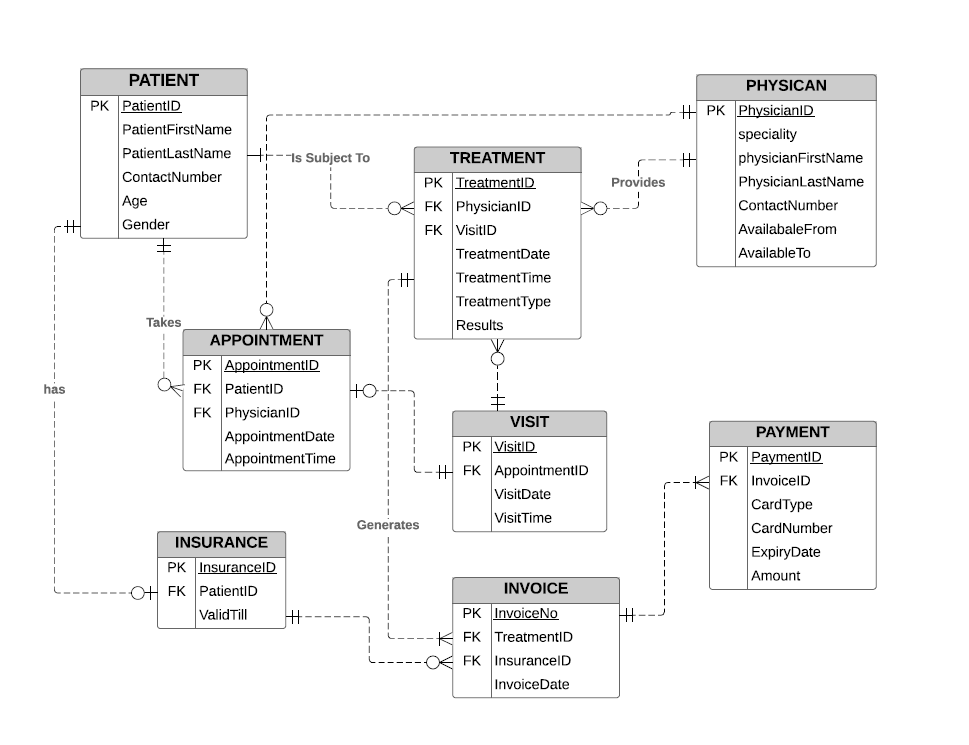
This table keeps track of the invoices and has a unique invoice #ID to uniquely distinguish every invoice. We have treatment #ID and insurance #ID to get the further details about that invoice.

* InvoiceID**(Primary Key)**
* AppointmentID **(Foreign Key)**
* InsuranceID **(Foreign Key)**
* InvoiceDate

**8.INSURANCE TABLE:**

This table keeps a track whether a patient has insurance coverage or not and whether the payment would be done via insurance. We have insurance #ID to uniquely identify each insurance along with the patient #ID to identify which patient has insurance or not.

* InsuranceID **(Primary Key)**
* PatientID **(Foreign Key)**
* ValidTill

**4.ENTITY RELATIONSHIP DIAGRAM :**

**5. RELATIONAL DATABASE SCHEMA**

**PATIENT TABLE:**

CREATE TABLE Patient

(PatientID INTEGER Not Null,

PatientFName VARCHAR (20) Not Null,

PatientLName VARCHAR (20) Not Null,

ContactNumber CHAR (10) Not Null,

Age INTEGER Not Null,

Gender VARCHAR (1) Not Null,

Primary Key (PatientID))

**PHYSICIAN TABLE:**

CREATE TABLE Physician

(PhysicianID INTEGER Not Null,

Speciality VARCHAR(50) Not Null,

PhysicianFName VARCHAR(20) Not Null,

PhysicianLName VARCHAR(20) Not Null,

ContactNumber INTEGER Not Null

Primary Key (PhysicianID))

**VISIT TABLE :**

CREATE TABLE Visit

(VisitID INTEGER Not Null,

VisitDate DateTime Null,

AppointmentID INTEGER Not Null,

Primary key (VisitID)

Foreign Key (AppointmentID) references Appointment)

**TREATMENT TABLE :**

CREATE TABLE Treatment

(TreatmentID INTEGER Not Null,

TreatmentDate DateTime Null,

TreatmentTime DateTime Null,

PatientID INTEGER Not Null,

PhysicianID INTEGER Not Null,

VisitID INTEGER Not Null,

TreatmentType Varchar (50) Not Null,

Results Varchar (50) Null

Primary Key (TreatmentID)

Foreign Key (PatientID) references Patient

Foreign Key (PhysicianID) references Physician

Foreign Key (VisitID) references Visit

**PAYMENT TABLE:**

CREATE TABLE Payment

(PaymentID INTEGER Not Null PRIMARY KEY,

CardType INTEGER Not Null,

CardNo INTEGER Not Null,

CardExpDate DATETIME Not Null,

PaymentAmount FLOAT Not Null,

PaymentTime DATETIME Not Null,

InvoiceID INTEGER Not Null,

Primary Key (PaymentID),

Foreign Key (InvoiceID) references Invoice)

**INSURANCE TABLE:**

CREATE TABLE Insurance

(InsuranceID INTEGER Not Null,

PatientID INTEGER Not Null,

ValidTill Char (10) Not Null,

Primary Key (InsuranceID),

Foreign Key (PatientID) references Patient)

**APPOINTMENT TABLE:**

CREATE TABLE Appointment

(AppointmentI INTEGER Not Null,

AppointmentDate DateTime Not Null,

AppointmentTime DateTime Not Null,

PatientID INTEGER Not Null,

Primary Key (AppointmentID),

Foreign Key (PatientID) references Patient)

**INVOICE TABLE :**

CREATE TABLE Invoice

(InvoiceID INTEGER Not Null,

InvoiceDate DateTime Not Null,

AppointmentID INTEGER Not Null,

InsuranceID INTEGER Not Null,

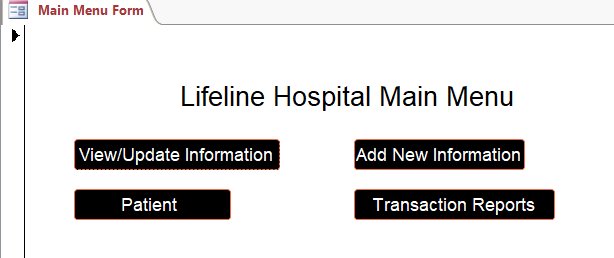
Primary Key (InvoiceID),

Foreign Key (InvoiceID) references Appointment,

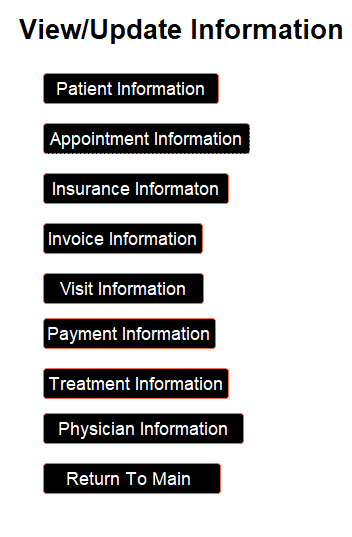
Foreign Key (InsuranceID) refernces Insurance

**6. MENU SCREEN FORM :**

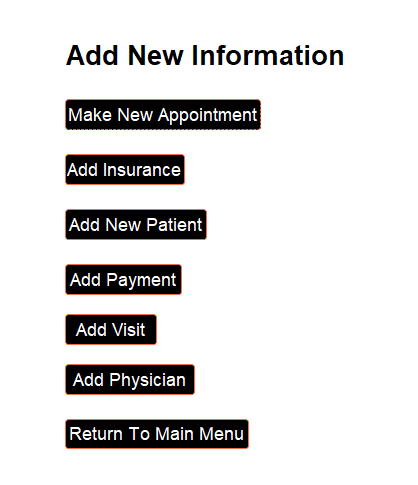
6.1 Lifeline Hospitals Main Menu :



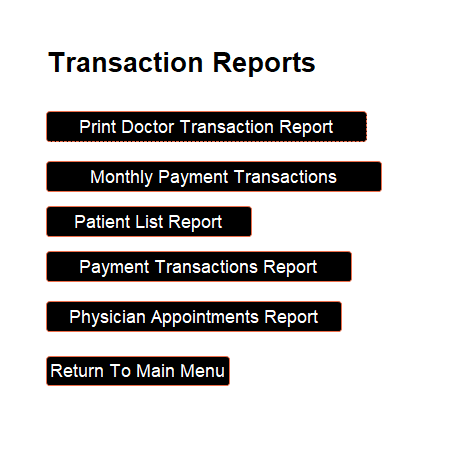
6.1.1 MAIN MENU SCREEN [After user clicks on ‘View/Update Information’ button, the below screen will appear, the users will be able to update/view the patient details.



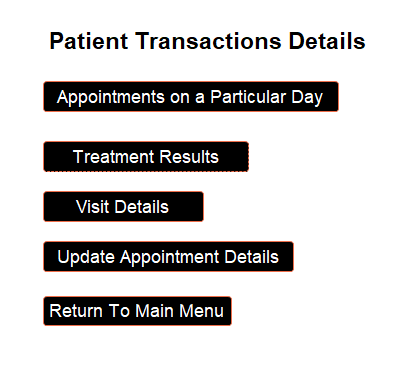
6.1.2 MAIN MENU SCREEN [After we click on ‘Add New Information’ the below screen will appear so that we can update the information]



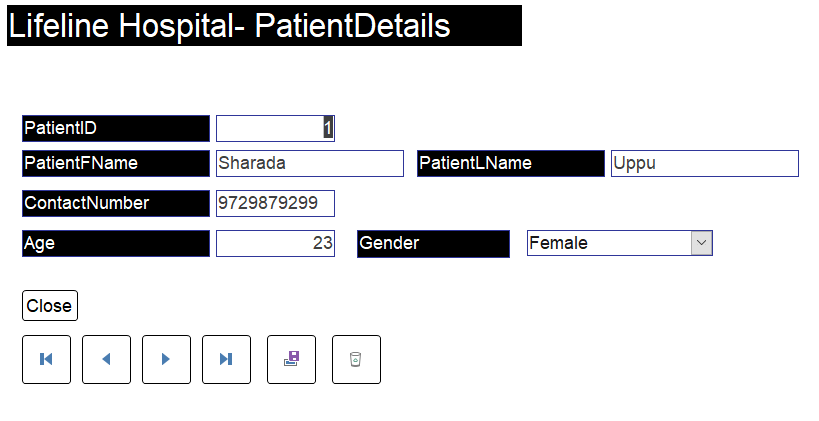
6.1.3 MAIN MENU SCREEN [After we click on ‘Transaction Reports’ the below screen will appear so that we can view the information from reports]



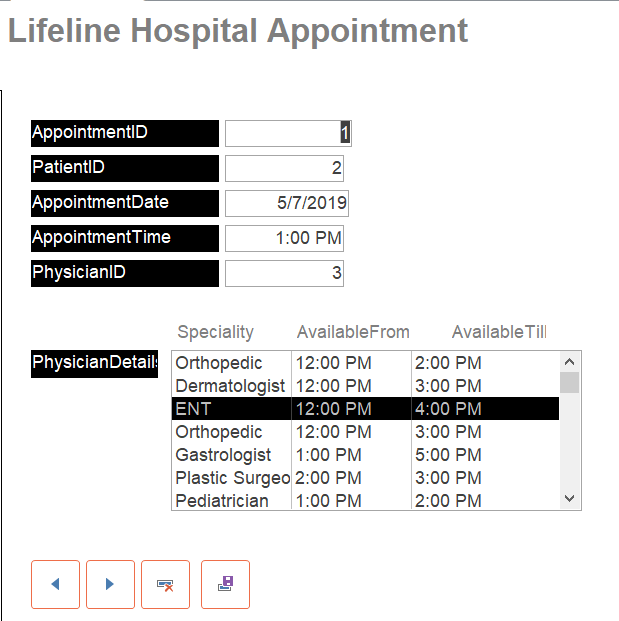
6.1.4 MAIN MENU SCREEN [After we click on ‘Patient Transaction’ the below screen will appear so that we can update/view the information]



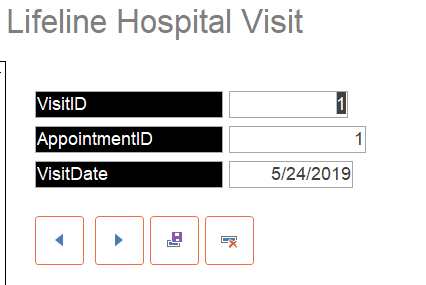
**6.2 Patient Information**



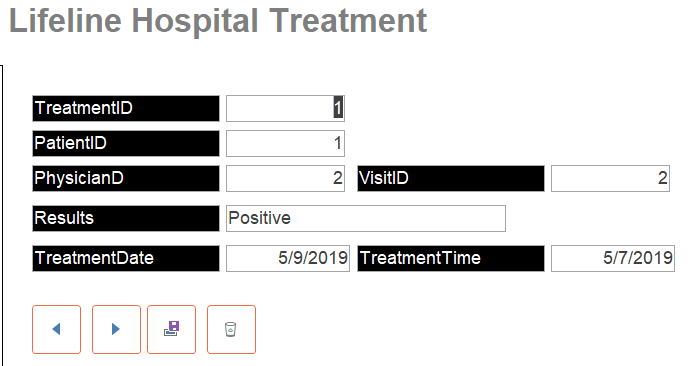
**6.3 APPOINTMENT DETAILS**



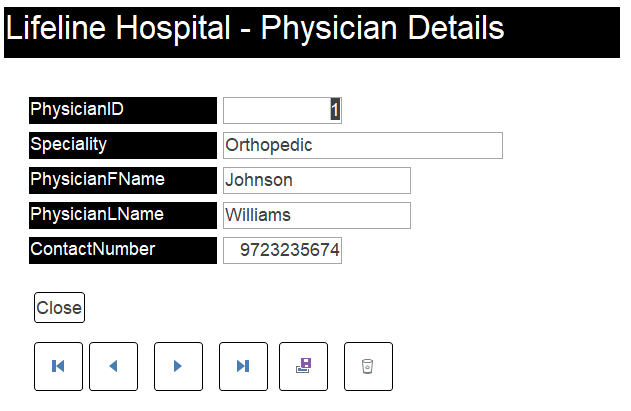
**6.3 Visit Information**



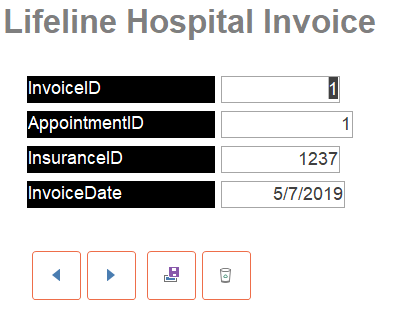
**6.4 Treatment Information**



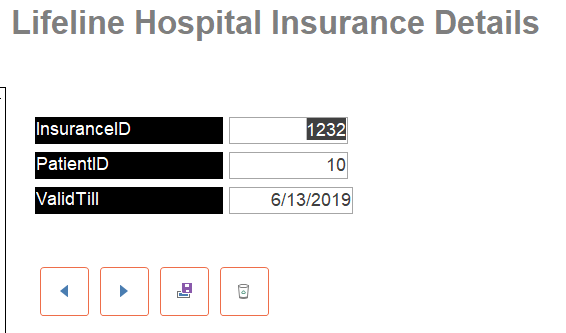
**6.5 Physician Information**



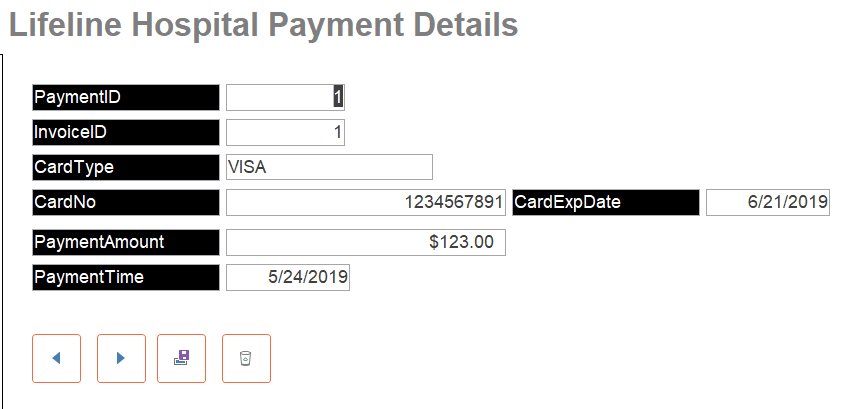
**6.6 Invoice Information**



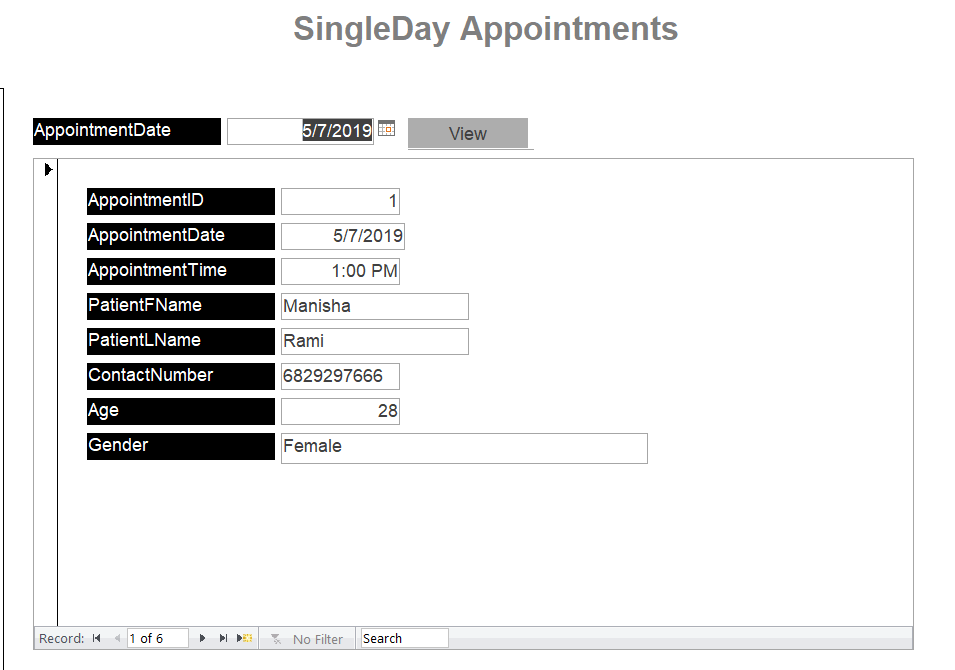
**6.7 Insurance Information**



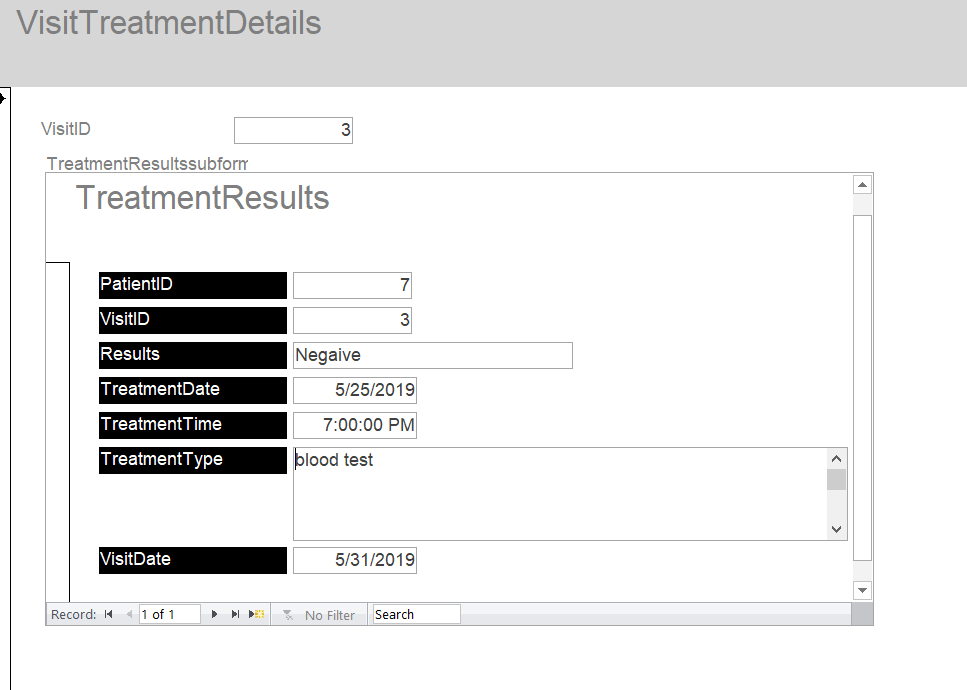
**6.8 Payment Information**



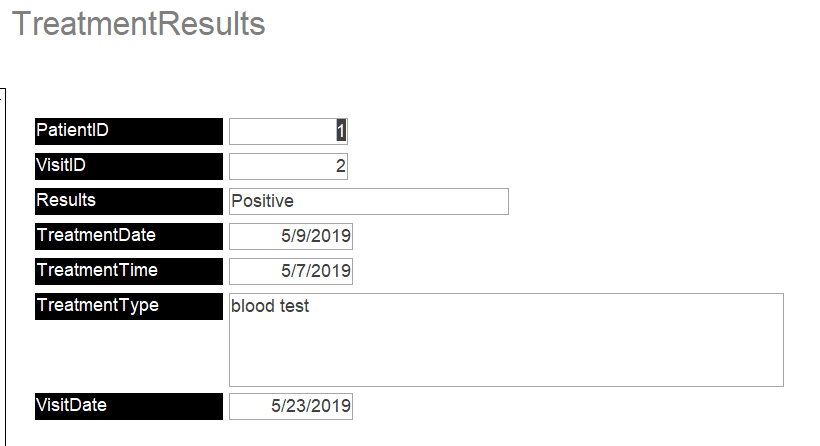
6.9.1 Patients Appointment Form [This Form shows all the Appointments for the day selected]



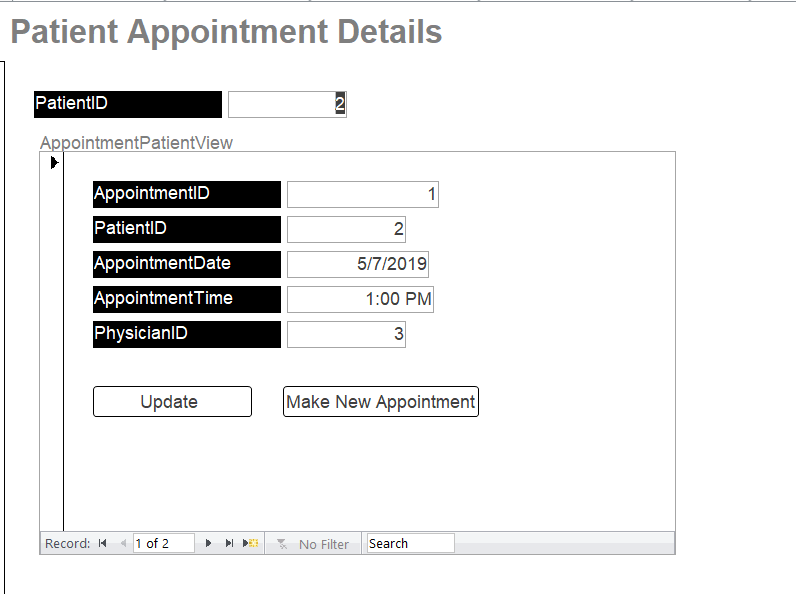
6.9.2 Visit Reports [Patient Treatment Results based on the visitID]



**6.9.3 View all patients Treatment Results**



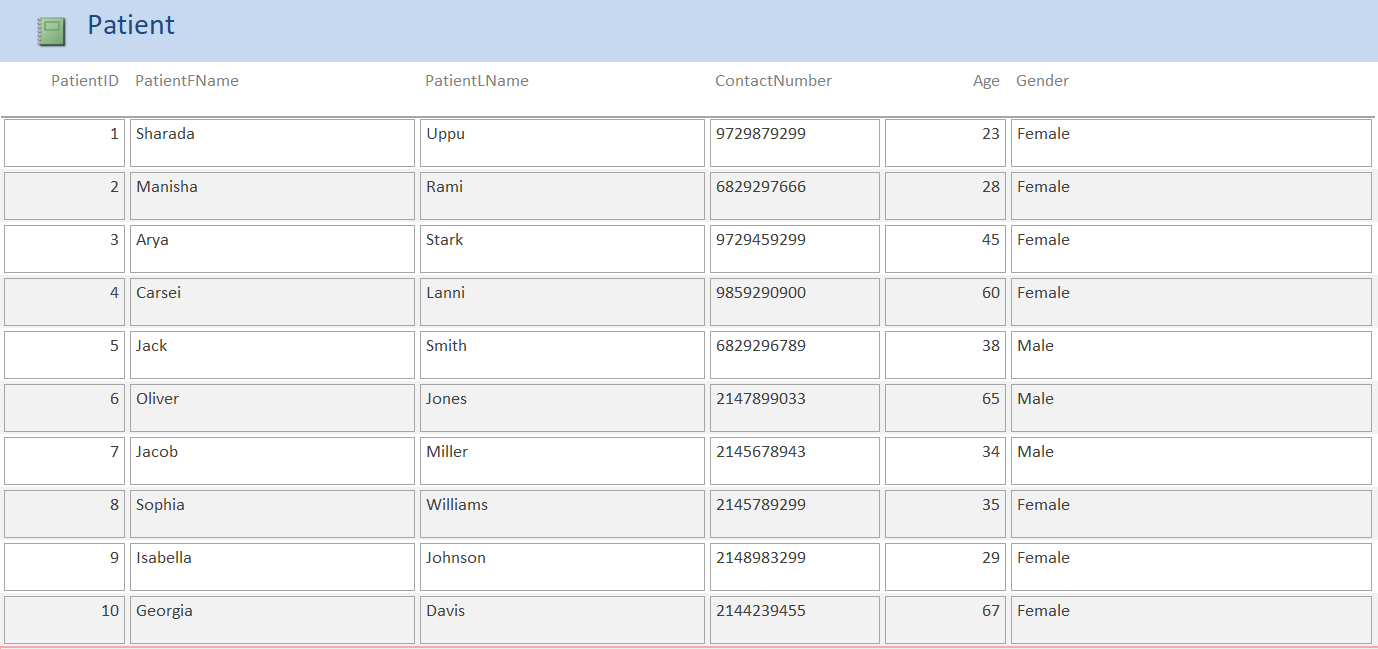
**6.9.4 Patient Appointment Details [view based on patientID and Update and can also make new appointment]**

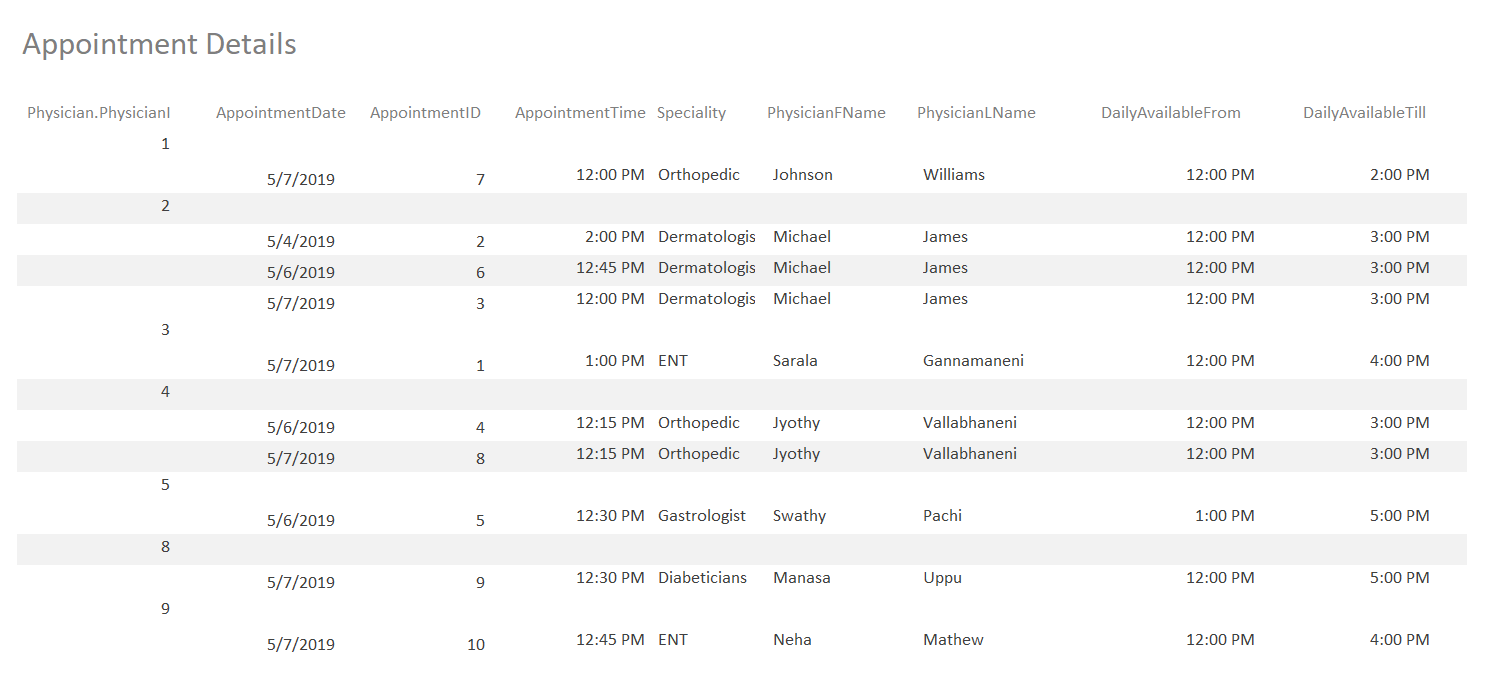


**SAMPLE REPORTS**

The following reports are provided as an example







The below Bar chart represents the total payment made during a month

|  |  |
| --- | --- |
| **Task Name** | **Team Member** |
| Problem Description | Pooja Bhamare |
| Organization Introduction | Pooja Bhamare |
| Scope of Database | Gowtham Phani Chandra Konda |
| Entity Relationship Diagram | Gowtham Phani Chandra Konda, Rajesh Kimidi |
| Relational Database Schema | Jagadesh Varma |
| Menu Screen Forms | Palle Sai Sushma |
| Data Input Screen Forms | Palle Sai Sushma |
| Queries | Jagadesh Varma |
| Forms | Pooja Bhamare, Rajesh Kimidi |
| Reports | Jagadesh Varma |
| Menus | Gowtham Phani Chandra Konda |
| Screen Captures | Rajesh Kimidi |
| Testing | Group |
| Documentation | Group |

Group Members Contributions