



CSC 621: Database Systems

Dr. Mary Krueger & Dr. Alan Nochenson

Final Report

By:

**Mohammed Alamoudi, Mohammed Al
Duniawi, Hemanth Nalamothu, Victor Logan**

Date: Dec/06/2016

Hospital Management System

Description:

The project that our group will be working on is a Hospital Management System, HMS. The System will be used by a hospital to keep the records of patients and manage the appointments for patients with doctors in the hospital and the cost of visits and operations for patients.

HMS will record and keep track of patients, appointments, bills, employees, login info, and department. Each patient will have a unique patient ID, First Name, Last name, Date of birth, Address, and Contact. Each appointment will have a unique Appointment ID, Appointment Date, Appointment Time, prescription, comments from doctor and appointment status. Each Bill will have a unique Bill ID, pending amount, Bill amount, and paid status. Each employee will have a unique employee ID, type of employment, image, First Name, Last Name, Address, Contact, Start Date and End Date.

There will be four users; The patient, the doctor, the receptionist, and the administrator. The patient can view and update their personal information. They can view bills and make a payment. The doctor can view and update their personal information. The doctor can see their list of appointments and reschedule if need be. The doctor can also make comments on appointments. The receptionist can register patients and create/edit appointments. The administrator can create and edit employees, doctors, and receptionists.

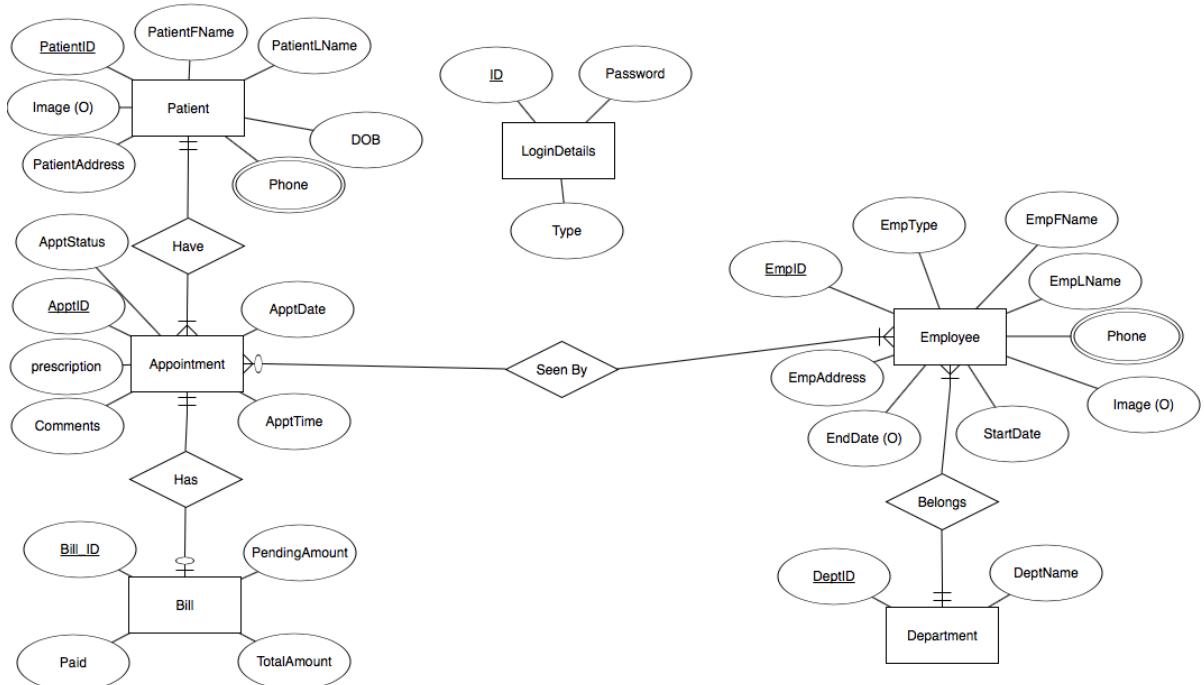
In terms of queries that will be supported, there will be create/insert, read/select, update and delete/remove. It will be Create New Patient, employee, Doctor, Receptionist, Admin, and new appointment. Read/display current appointments and view personal information. Update personal information, make payments, change appointments. Delete patient, appointment, doctor, and receptionist. In general, the Doctor will be able to Select, Update, and insert. The receptionist will be able to Insert, Select, Update, and Delete. The Admin will be able to Insert, Update, Delete, Create, and Select.

A patient can have one set of login details. A Patient can have one or many bills and a bill can only have one patient. An employee can be a doctor, a receptionist, or an admin. An employee can have one set of login details. A doctor can have one or many appointments. An appointment can be

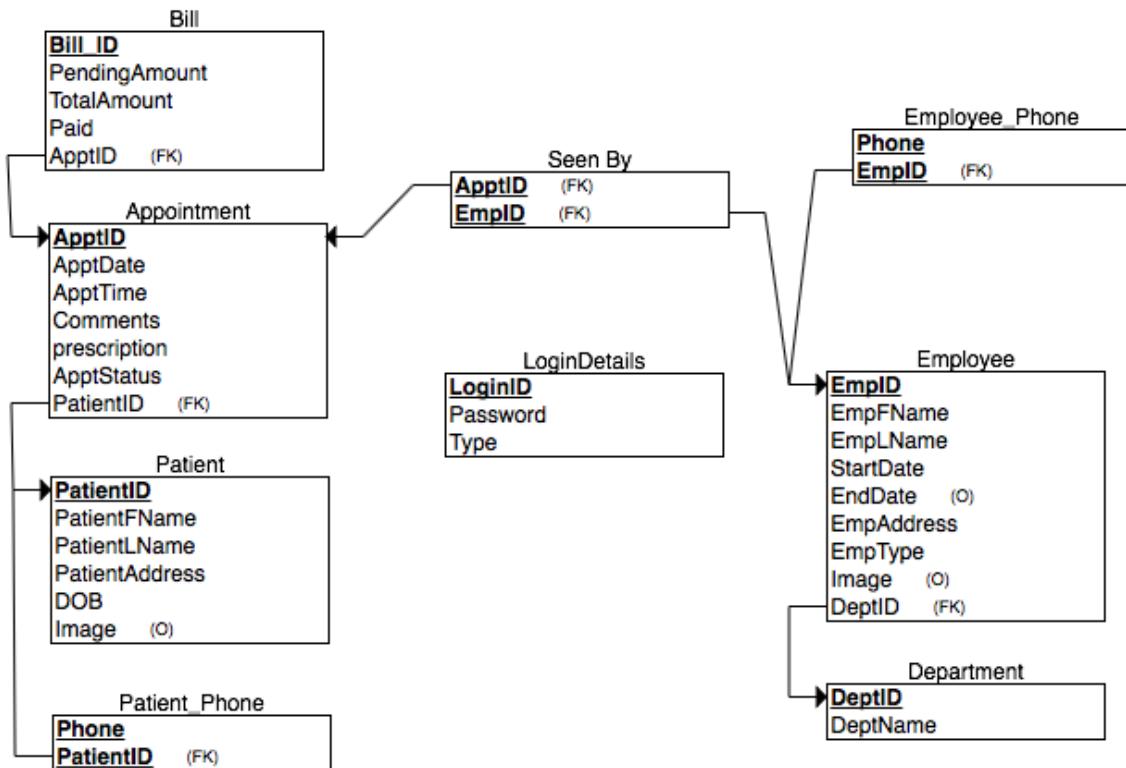
seen by one or many doctors. An appointment can have one bill and a bill can belong to one appointment.

A Delete Restriction is that a department record must not be deleted if its primary key value is referred to by a foreign key value. If an employee record is deleted from the Employee table, all foreign keys that refer to the primary key of this record will also be deleted. If a patient record is deleted from the patient table, all foreign keys that refer to the primary key of this record will also be deleted.

ER diagram:



Relational model:



Database Requirements:

HMS is a Hospital Management System that manages their Patients appointments bills and (*) Payments (*).

- HMS will keep track of its patients, Appointment, Bill, employees, Login details, department. (*) Transactions, Bill details and daily log (*)
- For each patient, HMS will keep track of unique patient ID, First Name, Last Name, Date of birth, Address, Contact, Image.
- For Each appointment, HMS will keep track of unique Appointment ID, Appointment Date, Appointment Time, prescription, Comments from Doctor and appointment status.
- For each Bill, HMS will keep track of unique Bill ID, Pending Amount, Bill Amount, Paid status and (*) multiple *Bill details* (*) .
- For each employee, HMS will keep track of unique Employee ID, type of Employment, Image, First Name, Last Name, Address, Contact, Start Date and End Date.
- A Patient can schedule one or many appointments. An Appointment should be for only one patient. HSM will NOT keep track of Patients with no appointment.
- For Each Login details, HMS will keep track of Login ID and Password.
- A Patient can have one Login details
- A Patient can have one or many bills. A bill will belong to one Patient.
- An employee can be doctor or Receptionist or Admin
- An employee can have one Login details
- A Doctor can take one or many appointments. An Appointment can be seen by only one or many Doctors.
- An appointment can have one bill. A Bill belong to one Appointment.

(*) Optional may add into the project if time favors.

Users:**Patients:**

- View/Update their personal information
- View Bills
- Make a Payment
- View / Cancel Appointment

Doctor:

- View/Update their personal information
- See the list of appointments.
- (*) Update Patient Status (*)
- Reschedule an appointment for patient
- Make comment/Note for appointment

Receptionist:

- Register new Patient
- Search for patient by:
 - Patient ID
 - First name
 - Last Name
 - Date of birth
 - Search between two dates of birth
- Make appointments
- Appointment search based on:
 - Patient's first name
 - Patient's last Name
 - Patient's date of birth
 - Appointment ID
 - Appointment date
 - Appointment time
 - Appointment status
 - Patient ID
 - Search between two appointment dates
- Update/delete appointment
- Update/delete patient

Admin:

- Create/Update/Delete Employees, Doctors, Receptionists
- Reset Passwords
- Doctor report based on
 - Month
 - Month and Year
 - Including employees not working currently
- Financial report based on
 - Month
 - Month and year

Queries:

- **Create**
 - Create new Patient, Employee, Doctor, Receptionist, Admin.
 - Create new Appointment.
- **Read**
 - Read/display current Appointments.
 - View personal information, Bills.
- **Update**

- Update personal information
 - Make Payments
 - Change appointment
- **Delete**
- Delete Patient, Appointment
 - Delete Doctor, Receptionist.

FD Closure:

In our project all the tables are in the 3rd normalization phase and the tables will be as:

Appointment

Column	Type	Can Be Null	Foreign Key to	Comments
<u>ApptID</u>	int(11)	no		
AppDate	date	no		
AppTime	time	no		
Comments	varchar(1000)	yes		
Prescription	varchar(500)	yes		
ApptStatus	varchar(25)	yes		
PatientID	int(11)	no	PatientID in Patient table	

Bill

Column	Type	Can Be Null	Foreign Key to	Comments
<u>BillID</u>	int(11)	no		
PendingAmount	decimal(10,2)	no		
TotalAmount	decimal(10,2)	no		
Paid	decimal(10,2)	no		
ApptID	int(11)	no	ApptID in Appointment table	

Department

Column	Type	Can Be Null	Foreign Key to	Comments
<u>DeptID</u>	int(11)	no		
DeptName	varchar(50)	no		

Employee

Column	Type	Can Be Null	Foreign Key to	Comments
<u>EmpID</u>	int(11)	no		
EmpFName	varchar(40)	no		
EmpLName	varchar(40)	no		
StartDate	date	no		
EndDate	date	yes		
EmpAddress	varchar(50)	no		
EmpType	varchar(15)	no		
Image	longblob	yes		
DeptID	int(11)	no	DeptID in Department table	

Employee_Phone

Column	Type	Can Be Null	Foreign Key to	Comments
<u>Phone</u>	varchar(15)	no		
<u>EmpID</u>	int(11)	no	EmpID in Employee table	

LoginDetails

Column	Type	Can Be Null	Foreign Key to	Comments
<u>LoginID</u>	int(11)	no		
Password	varchar(100)	no		
Type	varchar(15)	no		

Patient

Column	Type	Can Be Null	Foreign Key to	Comments
<u>PatientID</u>	int(11)	no		
PatientFName	varchar(25)	no		
PatientLName	varchar(25)	no		
PatientAddress	varchar(100)	yes		
DOB	date	no		

Image	longblob	yes		
-------	----------	-----	--	--

Patient_Phone

Column	Type	Can Be Null	Foreign Key to	Comments
<u>Phone</u>	varchar(15)	no		
<u>PatientID</u>	int(11)	no	PatientID in Patient table	

SeenBy

Column	Type	Can Be Null	Foreign Key to	Comments
<u>ApptID</u>	int(11)	no	ApptID in Appointment table	
<u>EmpId</u>	int(10)	no	EmpID in Employee table	

SQL for creation and population of the tables:

```
--  
-- Table structure for table `appointment`  
--  
  
CREATE TABLE `Appointment` (  
    `ApptID` int(11) NOT NULL,  
    `ApptDate` date NOT NULL,  
    `ApptTime` time NOT NULL,  
    `Comments` varchar(1000) DEFAULT NULL,  
    `Prescription` varchar(500) DEFAULT NULL,  
    `ApptStatus` varchar(400) DEFAULT NULL,  
    `PatientID` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
-- Dumping data for table `appointment`  
--  
  
INSERT INTO `Appointment` (`ApptID`, `ApptDate`, `ApptTime`, `Comments`, `Prescription`,  
    `ApptStatus`, `PatientID`) VALUES  
(5601, '2017-04-04', '09:30:00', NULL, NULL, 'closed', 198001),  
(5602, '2017-01-16', '11:34:00', 'needs to pay', NULL, 'closed', 198002),  
(5603, '2016-08-08', '08:36:00', NULL, 'placebo', 'closed', 198003),  
(5604, '2016-02-10', '14:50:00', 'May reschedule', '2 bottles', '12008', 198004),  
(5605, '2016-06-22', '11:32:00', NULL, '3.5 bottles', 'closed', 198005),  
(5606, '2017-02-02', '14:45:00', 'Child', 'none', '12017', 198006),  
(5607, '2017-02-08', '10:15:00', 'Needs to schedule another appointment', 'rest', '12018', 198007),  
(5608, '2016-05-31', '13:30:00', NULL, NULL, 'closed', 198008),
```

```
(5609, '2017-01-05', '09:15:00', 'Needs to schedule for child', '1 bottle', '12004', 198009),
(5610, '2016-11-22', '11:15:00', NULL, NULL, '12005', 198010),
(5611, '2016-10-11', '13:15:00', 'x-rays needed', NULL, 'closed', 198011),
(5612, '2017-01-12', '10:15:00', 'none', '4 bottles', '12011', 198012),
(5613, '2016-06-29', '14:35:00', 'is a child', 'Children''s tylenol', '12017', 198013),
(5614, '2016-10-24', '10:15:00', 'Needs blood work done', '2 bottles', 'closed', 198014),
(5615, '2016-09-20', '09:30:00', NULL, 'physical rehab', 'closed', 198015),
(5616, '2017-02-02', '13:45:00', 'needs to schedule a physical', NULL, 'closed', 198016),
(5617, '2016-05-17', '15:30:00', 'issues with payment', NULL, 'closed', 198017),
(5618, '2017-06-14', '15:30:00', NULL, NULL, '12005', 198018),
(5619, '2017-04-21', '13:29:00', 'Needs to pay', NULL, '12008', 198019),
(5620, '2017-07-13', '10:15:00', NULL, NULL, '12011', 198020);
```

```
--  
-- Table structure for table `bill`  
--
```

```
CREATE TABLE `Bill` (
  `BillId` int(11) NOT NULL,
  `PendingAmount` decimal(10,2) NOT NULL,
  `TotalAmount` decimal(10,2) NOT NULL,
  `Paid` decimal(10,2) NOT NULL,
  `ApptID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `bill`  
--
```

```
INSERT INTO `Bill` (`BillId`, `PendingAmount`, `TotalAmount`, `Paid`, `ApptID`) VALUES
(623001, '100.00', '1000.00', '900.00', 5601),
(623021, '50.00', '1000.00', '950.00', 5602),
(623022, '0.00', '10.00', '10.00', 5603),
(623023, '80.00', '100.00', '20.00', 5604),
(623024, '150.00', '300.00', '150.00', 5605),
(623025, '50.00', '100.00', '50.00', 5606),
(623026, '25.00', '50.00', '25.00', 5607),
(623027, '0.00', '100.00', '100.00', 5608),
(623028, '20.00', '40.00', '20.00', 5609),
(623029, '0.00', '750.00', '750.00', 5610),
(623030, '0.00', '700.00', '700.00', 5611),
(623031, '0.00', '600.00', '600.00', 5612),
(623032, '0.00', '730.00', '730.00', 5613),
(623033, '30.00', '90.00', '60.00', 5614),
(623034, '20.00', '80.00', '60.00', 5615),
(623035, '500.00', '1100.00', '600.00', 5616),
(623036, '25.00', '100.00', '75.00', 5617),
(623037, '20.00', '100.00', '80.00', 5618),
(623038, '75.00', '150.00', '75.00', 5619),
(623039, '0.00', '150.00', '150.00', 5620);
```

```

-- 
-- Table structure for table `department` 

CREATE TABLE `Department` (
  `DeptId` int(11) NOT NULL,
  `DeptName` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `department` 

INSERT INTO `Department` (`DeptId`, `DeptName`) VALUES
(1, 'Surgeon'),
(2, 'Accounts'),
(3, 'Administration');

-----


-- 
-- Table structure for table `employee` 

CREATE TABLE `Employee` (
  `EmpID` int(11) NOT NULL,
  `EmpFName` varchar(40) NOT NULL,
  `EmpLName` varchar(40) NOT NULL,
  `StartDate` date NOT NULL,
  `EndDate` date DEFAULT NULL,
  `EmpAddress` varchar(50) NOT NULL,
  `EmpType` varchar(15) NOT NULL,
  `Image` longblob,
  `DeptID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- 
-- Dumping data for table `employee` 

INSERT INTO `Employee` (`EmpID`, `EmpFName`, `EmpLName`, `StartDate`, `EndDate`, `EmpAddress`, `EmpType`, `Image`, `DeptID`) VALUES
(12001, 'Hemanth', 'Nalamothu', '2016-10-11', NULL, 'Penn', 'Doctor', NULL, 1),
(12002, 'Tom', 'Keen', '2016-09-13', NULL, 'New York', 'recep', NULL, 2),
(12003, 'John', 'Smith', '2016-08-16', NULL, '5600, City ave.', 'Admin', NULL, 3),
(12004, 'Jimmy', 'Carter', '2016-08-16', NULL, 'Lansdale avenue', 'Doctor', NULL, 1),
(12005, 'Monkey', 'Luffy', '2016-08-16', NULL, 'Grandline ave', 'Doctor', NULL, 1),
(12006, 'Roanoa', 'Zoro', '2015-04-15', '2016-08-09', 'New Jersey', 'Admin', NULL, 3),
(12007, 'Ryan', 'Howard', '2016-06-14', NULL, 'Philadelphia', 'recep', NULL, 2),
(12008, 'John', 'Jaskal', '2015-06-09', NULL, 'Jacksonville', 'Doctor', NULL, 1),
(12009, 'Michael', 'Johnson', '2016-01-04', '2016-12-01', 'Park Avenue', 'recep', NULL, 2),
(12010, 'Chris', 'Redfield', '2016-08-16', NULL, 'Jacksonville', 'Admin', NULL, 3),
(12011, 'will', 'Arnett', '2016-02-08', NULL, 'Kentucky', 'Doctor', NULL, 1),

```

```
(12012, 'Son', 'Goku', '2015-03-17', '2015-11-25', 'West City', 'recep', NULL, 2),
(12013, 'Jimmy', 'Neutron', '2016-03-06', NULL, 'Retroville', 'recep', NULL, 2),
(12014, 'Barack', 'Obama', '2015-02-01', '2016-12-01', 'Washington', 'Admin', NULL, 3),
(12015, 'Bill', 'Burr', '2016-02-08', NULL, 'Lancaster Ave', 'recep', NULL, 2),
(12016, 'Ryan', 'Reynolds', '2016-03-07', '2016-12-08', 'California', 'recep', NULL, 2),
(12017, 'Ronald', 'McDonald', '2015-12-16', NULL, 'New York', 'Doctor', NULL, 1),
(12018, 'Regina', 'King', '2015-09-21', NULL, 'Florida', 'Doctor', NULL, 1),
(12019, 'Kim', 'Possible', '2015-01-05', '2016-09-15', '5600, City ave.', 'Admin', NULL, 3),
(12020, 'John', 'Smith', '2016-08-16', NULL, '5600, City ave.', 'Admin', NULL, 3);
```

```
--  
-- Table structure for table `employee_phone`  
--
```

```
CREATE TABLE `Employee_Phone` (
  `Phone` varchar(15) NOT NULL,
  `EmpID` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `employee_phone`  
--
```

```
INSERT INTO `Employee_Phone` (`Phone`, `EmpID`) VALUES
('1009568348', 12010),
('1213456572', 12003),
('1234567890', 12002),
('2205683643', 12011),
('2348907685', 12019),
('2647692546', 12005),
('4280646386', 12009),
('4762107437', 12005),
('5439086574', 12017),
('5853551617', 12001),
('7094327654', 12013),
('7396738753', 12004),
('8096549876', 12014),
('8300348641', 12006),
('8569085423', 12018),
('9069872314', 12012),
('9081233243', 12016),
('9084532387', 12015),
('9094563214', 12020),
('9150275614', 12008),
('9351257642', 12007);
```

```
--  
-- Table structure for table `logindetails`  
--
```

```
CREATE TABLE `LoginDetails` (
  `ID` int(10) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `Type` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `logindetails`  
--
```

```
INSERT INTO `LoginDetails` (`ID`, `Password`, `Type`) VALUES
(12001, 'ba3abb2c0cb388f3cd4e77de3c78ff51', 'Doctor'),
(12002, 'f6bc0623a4ab517ae89db46f368c09c4', 'recep'),
(12003, 'd2a9aaedbe3616c7be11e07856c29e2a', 'Admin'),
(12004, 'dsjlkdhsoiyhlkefjlkio8732667786423a', 'Doctor'),
(12005, '897hjghgfty453ryguhf78i6u8', 'Doctor'),
(12006, 'kjhuhiug765786gyufv756578tg7ub78g', 'Admin'),
(12007, 'kjbkhjg78tgiujb76476t998iuo987908yhouih', 'recep'),
(12008, 'jkhg7647564567c764576t786', 'Doctor'),
(12009, '876iughkjbbyut76785674756456e56edjhfty7767', 'recep'),
(12010, 'kuhg87687678tgyvg7868hjkg7868', 'Admin'),
(198001, 'c6e53a7e82a4138b330b17c4a91267a1', 'Patient'),
(198002, 'c787979eokjllksdjfo8d97f9asudofij', 'Patient'),
(198003, 'hsjdlfkajskldfj92387r98324759iuou985798', 'Patient'),
(198004, '7868sdf87sad6f87sd afsd', 'Patient'),
(198005, '8698y5iuhjkh sd9uifg6238974', 'Patient'),
(198006, '78678hjbvjhkg875786tg778g78buhbg', 'Patient'),
(198007, '78678hjbvjhkg875786tg778g78buhbg', 'Patient'),
(198008, 'kjh876897678tgbv876897', 'Patient'),
(198009, '987hjkjk8768676hu0897hjk', 'Patient'),
(198010, 'kjhg78687678ijkgh87587tui', 'Patient');
```

```
--  
-- Table structure for table `patient`  
--
```

```
CREATE TABLE `Patient` (
  `PatientID` int(11) NOT NULL,
  `PatientFName` varchar(25) NOT NULL,
  `PatientLName` varchar(25) NOT NULL,
  `PatientAddress` varchar(100) DEFAULT NULL,
  `DOB` date NOT NULL,
  `Image` longblob
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `patient`  
--
```

```
INSERT INTO `Patient` (`PatientID`, `PatientFName`, `PatientLName`, `PatientAddress`, `DOB`,
`Image`) VALUES
(198001, 'Mohammed', 'Alamoudi', 'Lancaster Ave. , Philadelphia, Pa', '1990-12-11', NULL),
```

```
(198002, 'Victor', 'Logan', 'City Ave. , Philadelphia, Pa', '1994-03-11', NULL),
(198003, 'Jack', 'Johnson', 'Park ave, Pleasantville, Pa', '1997-09-10', NULL),
(198004, 'Jimmy', 'Carter', 'lapsely lane , Philadelphia, Pa', '1980-03-11', NULL),
(198005, 'Joel', 'Velez', 'Pierce Street , Philadelphia, Pa', '1995-02-11', NULL),
(198006, 'Emmanuel', 'Johnson', 'Pierce St. , Philadelphia, Pa', '1989-06-23', NULL),
(198007, 'Jimmy', 'Johnson', 'New York', '1991-05-16', NULL),
(198008, 'Jess', 'Carsel', 'Kentucky', '1989-08-08', NULL),
(198009, 'Mary', 'Krueger', 'Pennsylvania', '1991-04-18', NULL),
(198010, 'jennifer', 'Aniston', 'Pennsylvania ', '1989-04-09', NULL),
(198011, 'Sarah', 'Jones', 'New Jersey', '1991-08-13', NULL),
(198012, 'Will', 'Johnson', 'Massachusetts ', '1989-05-16', NULL),
(198013, 'Matt', 'Patt', 'Texas', '1993-04-21', NULL),
(198014, 'KYLE', 'Chambley', 'Pennsylvania ', '1989-10-18', NULL),
(198015, 'Gio', 'Vencenzo', 'New Jersey', '1991-04-16', NULL),
(198016, 'Bruce', 'Springsteen', 'California', '1989-06-06', NULL),
(198017, 'Jeff', 'Bakular', 'Pennsylvania', '1992-05-19', NULL),
(198018, 'Colonel', 'Sanders', 'Florida', '1989-11-21', NULL),
(198019, 'John', 'Richards', 'Pennsylvania ', '1989-01-19', NULL),
(198020, 'kim', 'Possible', 'New Jersey', '1990-07-23', NULL);
```

```
--  
-- Table structure for table `patient_phone`
```

```
--  
CREATE TABLE `Patient_Phone` (  
    `Phone` char(15) NOT NULL,  
    `PatientID` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `patient_phone`
```

```
--  
INSERT INTO `Patient_Phone` (`Phone`, `PatientID`) VALUES  
(`0500111405`, 198001),  
(`2156099039`, 198002),  
(`9061438546`, 198003),  
(`6450982314`, 198004),  
(`7094386754`, 198005),  
(`7094239876`, 198006),  
(`5762097865`, 198007),  
(`4397687456`, 198008),  
(`9084897865`, 198009),  
(`3908763409`, 198010),  
(`6459086754`, 198011),  
(`6472340987`, 198012),  
(`7589072345`, 198013),  
(`7684569567`, 198014),  
(`7684989087`, 198015),  
(`6547894567`, 198016),  
(`7560983456`, 198017),  
(`6785430989`, 198018),
```

```
('8794560789', 198019),
('8795766987', 198020);
```

```
--  
-- Table structure for table `seenby`  
--
```

```
CREATE TABLE `SeenBy` (  
    `ApptID` int(11) NOT NULL,  
    `EmpId` int(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Dumping data for table `seenby`  
--
```

```
INSERT INTO `SeenBy` (`ApptID`, `EmpId`) VALUES  
(5601, 12001),  
(5602, 12004),  
(5603, 12005),  
(5604, 12008),  
(5605, 12011),  
(5606, 12017),  
(5607, 12018),  
(5608, 12001),  
(5609, 12004),  
(5610, 12005),  
(5611, 12008),  
(5612, 12011),  
(5613, 12017),  
(5614, 12018),  
(5615, 12001),  
(5616, 12001),  
(5617, 12004),  
(5618, 12005),  
(5619, 12008),  
(5620, 12011);
```

```
--  
-- Indexes for dumped tables  
--
```

```
--  
-- Indexes for table `appointment`  
--
```

```
ALTER TABLE `Appointment`  
    ADD PRIMARY KEY (`ApptID`),  
    ADD KEY `PatientID` (`PatientID`);
```

```
--  
-- Indexes for table `bill`  
--
```

```
ALTER TABLE `Bill`  
ADD PRIMARY KEY (`BillId`),  
ADD KEY `ApptID` (`ApptID`);  
  
--  
-- Indexes for table `department`  
--  
ALTER TABLE `Department`  
ADD PRIMARY KEY (`DeptId`);  
  
--  
-- Indexes for table `employee`  
--  
ALTER TABLE `Employee`  
ADD PRIMARY KEY (`EmpID`),  
ADD KEY `DeptID` (`DeptID`);  
  
--  
-- Indexes for table `employee_phone`  
--  
ALTER TABLE `Employee_Phone`  
ADD PRIMARY KEY (`Phone`,`EmpID`),  
ADD KEY `EmpID` (`EmpID`);  
  
--  
-- Indexes for table `logindetails`  
--  
ALTER TABLE `LoginDetails`  
ADD PRIMARY KEY (`ID`);  
  
--  
-- Indexes for table `patient`  
--  
ALTER TABLE `Patient`  
ADD PRIMARY KEY (`PatientID`);  
  
--  
-- Indexes for table `patient_phone`  
--  
ALTER TABLE `Patient_Phone`  
ADD PRIMARY KEY (`PatientID`,`Phone`);  
  
--  
-- Indexes for table `seenby`  
--  
ALTER TABLE `SeenBy`  
ADD PRIMARY KEY (`ApptID`,`EmpId`),  
ADD KEY `EmpID` (`EmpId`);  
  
--  
-- AUTO_INCREMENT for dumped tables  
--  
--
```

```

-- AUTO_INCREMENT for table `appointment`
--
ALTER TABLE `Appointment`
  MODIFY `ApptID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5621;
--
-- AUTO_INCREMENT for table `bill`
--
ALTER TABLE `Bill`
  MODIFY `BillId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=623040;
--
-- AUTO_INCREMENT for table `department`
--
ALTER TABLE `Department`
  MODIFY `DeptId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT for table `employee`
--
ALTER TABLE `Employee`
  MODIFY `EmpID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12021;
--
-- AUTO_INCREMENT for table `patient`
--
ALTER TABLE `Patient`
  MODIFY `PatientID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=198021;
--
-- Constraints for dumped tables
--

-- Constraints for table `appointment`
--
ALTER TABLE `Appointment`
  ADD CONSTRAINT `appointment_ibfk_1` FOREIGN KEY (`PatientID`) REFERENCES `Patient`(`PatientID`) ON DELETE CASCADE;

-- Constraints for table `bill`
--
ALTER TABLE `Bill`
  ADD CONSTRAINT `bill_ibfk_1` FOREIGN KEY (`ApptID`) REFERENCES `Appointment`(`ApptID`) ON DELETE CASCADE;

-- Constraints for table `employee`
--
ALTER TABLE `Employee`
  ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`DeptID`) REFERENCES `Department`(`DeptId`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `employee_phone`
--
ALTER TABLE `Employee_Phone`

```

```

ADD CONSTRAINT `employee_phone_ibfk_1` FOREIGN KEY (`EmpID`) REFERENCES
`Employee` (`EmpID`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `patient_phone`
ALTER TABLE `Patient_Phone`
ADD CONSTRAINT `patient_phone_ibfk_1` FOREIGN KEY (`PatientID`) REFERENCES
`Patient` (`PatientID`) ON DELETE CASCADE ON UPDATE CASCADE;

-- Constraints for table `seenby`
ALTER TABLE `SeenBy`
ADD CONSTRAINT `seenby_ibfk_1` FOREIGN KEY (`EmpId`) REFERENCES `Employee`(`EmpID`)
ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `seenby_ibfk_2` FOREIGN KEY (`ApptID`) REFERENCES `Appointment`(`ApptID`)
ON DELETE CASCADE ON UPDATE CASCADE;

```

Discuss how you anticipate the application to be used and how this drove your design:

This application is developed to be used by specific hospital for organizing appointments and keep track of the cost of visits and operations for patients. This application is designed to be used by four types of users: patient, doctor, receptionist, and the administrator

Interfaces:

Log In:

SIGN IN

[Sign Up Here](#)

Sign Up:

[Sign In Here](#)

Receptionist Interface:

New Patient:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Patient search:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient ID, name or date of birth:
 First Name Last Name mm/dd/yyyy

Search between two dates:
 mm/dd/yyyy mm/dd/yyyy

Patient search result:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient ID, name or date of birth:
 Patient ID First Name Last Name mm/dd/yyyy

Search between two dates:
 mm/dd/yyyy mm/dd/yyyy

PatientID	First Name	Last Name	PatientAddress	Birth of date	Edit	Delete	New Appointment
198001	Mohammed	Alamoudi	Lancaster	1990-12-11	Edit	Delete	New

Edit patient search result:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out



First Name: Last Name:

Address: Date of birth:

Image: No file chosen

Add a new appointment:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Add a new appointment:

Patient ID:

Appointment Date: mm/dd/yyyy

Appointment Time:

Comments:

ApptStatus:

Appointment search:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient's ID, name, or date of birth:

Search by appointment ID, date, time, status, or Patient Id:

Search between two dates:

Search for Jimmy:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient's ID, name, or date of birth:

Search by appointment ID, date, time, status, or Patient Id:

Search between two dates:

Appointment ID	Patient ID	Date	Time	Comments	Prescription	Status	Edit	Delete
5604	198004	2016-02-10	14:50:00	May reschedule	2 bottles		Edit	Delete
5607	198007	2017-02-08	10:15:00	Needs to schedule another appointment	rest		Edit	Delete

Edit appointment:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Patient ID:

Appointment Date:

Appointment Time:

Comments:

ApptStatus:

Search for appointment between two dates (12-05-2016 and 02-20-2017):

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient's ID, name, or date of birth:

Search by appointment ID, date, time, status, or Patient Id:

Search between two dates:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out

Search by patient's ID, name, or date of birth:

Search by appointment ID, date, time, status, or Patient Id:

Search between two dates:

Appointment ID	Patient ID	Date	Time	Comments	Prescription	Status	Edit	Delete
S602	198002	2017-01-16	11:34:00	needs to pay		closed	Edit	Delete
S607	198007	2017-02-08	10:15:00	Needs to schedule another appointment		rest	Edit	Delete
S609	198009	2017-01-05	09:15:00	Needs to schedule for child		1 bottle	Edit	Delete
S612	198012	2017-01-12	10:15:00	none		4 bottles	Edit	Delete
S616	198016	2017-02-02	13:45:00	needs to schedule a physical		closed	Edit	Delete

Edit receptionist profile:

New Patient Patient Search New Appointment Appointment Search Welcome Tom Edit Profile Sign Out



First Name:

Last Name:

Address:

Image: No file chosen

Doctor Interface

Home

Home Appointments

Welcome Hemanth Edit Profile Sign Out

Waiting List

Appointment ID	PatientID	First Name	Last Name	Age	Select
5615	198015	Gio	Vencenzo	25	Select

Selected Patients

No Patient Selected

When a patient is Selected

Home Appointments

Welcome Hemanth

Edit Profile

Sign Out

Waiting List

No appointments found

Selected Patients

Appointment ID	PatientID	First Name	Last Name	Age	UpdateStatus	Comment	Submit
5615	198015	Gio	Vencenzo	25	--Select--	Be careful to avoid taking too much	<input type="button" value="Update"/> <input type="button" value="undo select"/>

UpdateStatus DropDown

Selected Patients

Appointment ID	PatientID	First Name	Last Name	Age	UpdateStatus	Comment	Submit
5615	198015	Gio	Vencenzo	25	<input checked="" type="button" value="--Select--"/> <input type="button" value="close"/> <input type="button" value="Appointment"/> <input type="button" value="Reschedule"/>	Be careful to avoid taking too much	<input type="button" value="Update"/> <input type="button" value="undo select"/>

--Reschedule

Done Appointments

Welcome Hemanth Edit Profile Sign Out

Waiting List

Reschedule

Date:

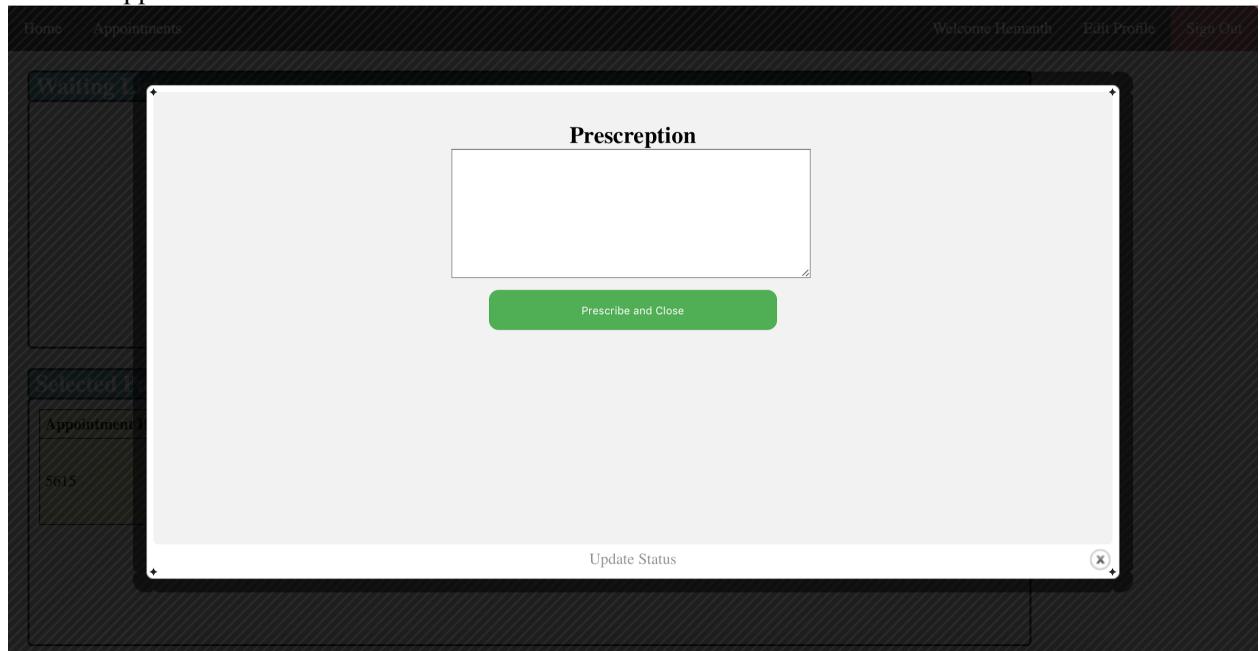
Time:

Selected Patients

Appointment
5615

Update Status

--Close Appointment



Appointments DropDown

A screenshot of a medical software interface. At the top, there's a navigation bar with links for Home, Appointments, Welcome Hemanth, Edit Profile, and Sign Out. Below the navigation bar, there's a search bar with fields for 'Search Appointments' and 'View Appointments'. A dropdown menu titled 'Waiting List' is open, showing a table of appointments. The table has columns for Appointment ID, PatientID, First Name, Last Name, Age, and Select. One row is visible, showing Appointment ID 5615, PatientID 198015, First Name Gio, Last Name Vencenzo, Age 25, and a 'Select' button. Below the dropdown, there's a section titled 'Selected Patients' which displays the message 'No Patient Selected'. At the bottom left, there's a note: 'Open "www.malamodi.com/internetApp/doctor/view.php" in a new tab'.

Appointment ID	PatientID	First Name	Last Name	Age	Select
5615	198015	Gio	Vencenzo	25	Select

Search Appointment(With year)

[Home](#) [Appointments](#)

Welcome Hemanth [Edit Profile](#) [Sign Out](#)

Search Appointments :

Month :

Date :

Year :

 [Advanced Search](#)

Appointments on 2016.

Previous Appointments

Appt ID	PatientID	First Name	Last Name	Age	Date	Prescription	Comment	Update	Reopen
5615	198015	Gio	Vencenzo	25	2016-12-05	Acetaminophen (Tylenol, others) or ibuprofen (Advil, Motrin IB, others).	Be careful to avoid taking too much	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>
5625	198023	John	Adams	26	2016-12-05		You should not take zolpidem unless you are able to get a full night's sleep -- at least 7 to 8 hours.	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>
5609	198009	Frank	Green	27	2016-05-21	single dose Diflucan	This drug kills fungus and yeast throughout your body.	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>

Search Appointment(With month and year)

[Home](#) [Appointments](#)

Welcome Hemanth [Edit Profile](#) [Sign Out](#)

Search Appointments :

Month :

Date :

Year :

 [Advanced Search](#)

Appointments on December, 2016.

Previous Appointments

Appt ID	PatientID	First Name	Last Name	Age	Date	Prescription	Comment	Update	Reopen
5625	198023	John	Adams	26	2016-12-05		You should not take zolpidem unless you are able to get a full night's sleep -- at least 7 to 8 hours.	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>
5615	198015	Gio	Vencenzo	25	2016-12-05	Acetaminophen (Tylenol, others) or ibuprofen (Advil, Motrin IB, others).	Be careful to avoid taking too much	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>

Search Appointment(With date, month and year)

[Home](#) [Appointments](#)

Welcome Hemanth [Edit Profile](#) [Sign Out](#)

Search Appointments :

Month :

Date :

Year :

 [Advanced Search](#)

Appointments on May, 31, 2016.

Previous Appointments

Appt ID	PatientID	First Name	Last Name	Age	Date	Prescription	Comment	Update	Reopen
5608	198008	Jess	Carsel	27	2016-05-31	single dose Diflucan	This drug kills fungus and yeast throughout your body, so you may have minor side effects, such as stomach	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>

Advance Search(With first name: gio)

[Home](#) [Appointments](#)

Welcome Hemanth [Edit Profile](#) [Sign Out](#)

Advance Search Appointments :

Patient ID:

First Name:

Last Name:

Appointments on First Name: Gio.

Previous Appointments

Appt ID	PatientID	First Name	Last Name	Age	Date	Prescription	Comment	Update	Reopen
5615	198015	Gio	Vencenzo	25	2016-12-05	Acetaminophen (Tylenol, others) or ibuprofen (Advil, Motrin IB, others).	Be careful to avoid taking too much	<input type="button" value="Update"/>	<input type="button" value="ReOpen"/>

Update profile(when submit button clicked)

Home Appointments

Welcome Hemanth Edit Profile Sign Out

Your profile has been successfully updated

You will be redirected in 5 second(s).



First Name: Hemanth

Last Name: Nalamothu

Address: Penn

Image: Choose File no file selected

Submit

Admin Interface

Home

Home Report User

Welcome John Edit Profile Sign Out

Report DropDown

Home Report User

Welcome John Edit Profile Sign Out

Doctor Report

Billing Report

--Doctor Report(By year)

Home Report User

Welcome John Edit Profile Sign Out

Doctor Report :

Month :

Year :

Search only Current employee.

Submit

Doctor Report for 2016.

Employee ID	First Name	Last Name	Department	Number of patients seen
12001	Hemanth	Nalamothu	Surgeon	5
12005	Monkey	Luffy	Surgeon	2
12017	Ronald	Mcdonald	Surgeon	2
12008	John	Jaskal	Surgeon	2
12004	Jimmy	Carter	Surgeon	1
12018	Regina	King	Surgeon	1
12011	will	Arnett	Surgeon	1

--Doctor Report(By month and year)

Home Report User

Welcome John Edit Profile Sign Out

Doctor Report :

Month :

Year :

Search only Current employee.

Submit

Doctor Report for October, 2016.

Employee ID	First Name	Last Name	Department	Number of patients seen
12008	John	Jaskal	Surgeon	1
12018	Regina	King	Surgeon	1
12004	Jimmy	Carter	Surgeon	0
12011	will	Arnett	Surgeon	0
12005	Monkey	Luffy	Surgeon	0
12001	Hemanth	Nalamothu	Surgeon	0
12017	Ronald	Mcdonald	Surgeon	0

User DropDown

Home Report User

Welcome John Edit Profile Sign Out

Add new User

Update/delete User

--Update/delete User

Home Report User Welcome John Edit Profile Sign Out

Search by ID, First Name or Last Name:

Login ID:

First Name:

Last Name:

Type: Employee Patient

Search

--Update/delete User(Search with first name: john)

Home Report User Welcome John Edit Profile Sign Out

Search by ID, First Name or Last Name:

Login ID:

First Name:

Last Name:

Type: Employee Patient

Search

Employee search Results:

Employee ID	First Name	Last Name	Employee Address	Reset Password	Delete
12003	John	Smith	5600, City ave.	Reset Password	Delete
12008	John	Jaskal	Jacksonville	Reset Password	Delete

--Add new user

Home Report User Welcome John Edit Profile Sign Out

Add user

First Name:

Last Name:

Address:

User Type: 

Department: 

Edit Profile

Home Report User Welcome John Edit Profile Sign Out



First Name:

Last Name:

Address:

Image: no file selected

Patient Interface

Patient Home Page

Home My Appointments My Bills Welcome John Update My profile My Profile Sign Out

Patient - Home Page

Viewing my appointments

[Home](#) [My Appointments](#) [My Bills](#)

Welcome **John** [Update My profile](#) [My Profile](#) [Sign Out](#)

My Appointments

Appointment ID	Appointment Date	Time	Status	Doctor	Department	Comments	Prescription	CANCEL
5630	2016-10-01	09:30:00	closed	Hemanth Nalamothu	Surgeon			<button>Cancel</button>
5635	2016-10-10	10:00:00	closed	Hemanth Nalamothu	Surgeon			<button>Cancel</button>
5631	2016-10-02	10:00:00	closed	Jimmy Carter	Surgeon			<button>Cancel</button>
5633	2016-10-06	10:00:00	closed	Jimmy Carter	Surgeon			<button>Cancel</button>
5636	2016-11-09	11:00:00	closed	Monkey Luffy	Surgeon			<button>Cancel</button>

Viewing My bills

[Home](#) [My Appointments](#) [My Bills](#)

Welcome **John** [Update My profile](#) [My Profile](#) [Sign Out](#)

My Bills

ID	Appointment Date	Time	Doctor	Department	Bill ID	Pending Amount	Total Amount	Paid
5630	2016-10-01	09:30:00	Hemanth Nalamothu	Surgeon	623041	100.00	150.00	50.00
5635	2016-10-10	10:00:00	Hemanth Nalamothu	Surgeon	623045	60.00	260.00	200.00
5631	2016-10-02	10:00:00	Jimmy Carter	Surgeon	623042	50.00	300.00	250.00
5633	2016-10-06	10:00:00	Jimmy Carter	Surgeon	623044	0.00	100.00	100.00
5636	2016-11-09	11:00:00	Monkey Luffy	Surgeon	623046	200.00	310.00	110.00

-Viewing my profile

Home	My Appointments	My Bills	Welcome John	Update My profile	My Profile	Sign Out
------	-----------------	----------	--------------	-------------------	------------	----------

My Profile

	
Patient ID:	198024
First Name:	John
Last Name:	Adams
Date of Birth:	1990-01-10
Address:	6100 City Ave
Phone 1 :	215-233-4567
Phone 2 :	215-343-6677

-Updating My Profile

Home	My Appointments	My Bills	Welcome John	Update My profile	My Profile	Sign Out
------	-----------------	----------	--------------	-------------------	------------	----------

Update My Profile

John
Adams
1/10/1990
6100 City Ave

215-233-4567
215-343-6677
UPDATE PROFILE

- **Discuss how you implement constraints**
 - We used unique keys for each table to identify and for mapping relationship between tables
 - We used foreign keys to populate tables
 - For each user we used a userType to identify the type of user and to redirect to the respective pages.
 - We used the doctor ID as appointment status to redirect patient to respective doctor

- **Discuss how you ensure integrity**
 - We have four kind of users each with a specific privileges.
 - Everyone within a table has an unique id
 - If a patient id matches an employee id they will not be able to access information due to being in different tables and the user type attribute is different.
 - We hashed password in the LoginDetails table.

- **Discuss updates that will require transactions be built**
 - we used the transaction while a doctor selecting an appointment so that no appointments get selected/updated twice(by two doctors).

- **Discuss error checking - non sensical data entered on form**
 - In the registration page, the first name, last name, date of birth, address, one phone number, and password are requirements but not the second phone number or the image.
 - In the receptionist interface, adding a new appointment will have to fill the patient id, appointment date, appointment time but not the comments and appointment status.
 - In Doctor interface:
 - When updating profile we made sure all the required input values are not empty and ensured the format of profile image upload is actually an image before updating
 - when rescheduling an appointment we validate the form for empty values and also the date(No date before current date) of the new appointment.
 - When searching/Advance search for appointment we check for empty input values and also date format.
 - We also added popup for confirmation before reopening a closed appointment.
 - In Admin interface:
 - When updating profile we make sure all the required input values are not empty and ensured the format of profile image upload is actually an image before updating
 - When searching for users we check for empty input values
 - When creating a new user we make sure all the values are entered correctly and all the required options are selected
 - In Recipeniest interface:
 - when creating a new patient we make sure all the required fields are filled.
 - when adding a new appointment we make sure all the required fields are filled with correct values and not the optional ones.
 - In Patient Interface:
 - In update profile page, the patient can not submit empty form or fields with empty value

identify 3 indexes other than primary key:

- PatientFName and PatientLName for Patient table
- EmpFName and EmpLName for Employee

Table for assigned pieces:

Peace Assigned:	Names:	Due Date:
ER Diagram	<ul style="list-style-type: none"> • Hemnthy Nalamothu • Mohammed Alamoudi • Mohammed Alduniawi 	Nov/6/2016
Relational Model	<ul style="list-style-type: none"> • Hemnthy Nalamothu • Mohammed Alamoudi • Mohammed Alduniawi 	Nov 10, 2016
Reviewing ER Diagram	<ul style="list-style-type: none"> • Victor Logan • Mohammed Alduniawi 	Nov 7, 2016
Reviewing Relational Model	<ul style="list-style-type: none"> • Victor Logan 	Nov 11, 2016
Create Database & Tables	<ul style="list-style-type: none"> • Mohammed Alduniawi 	Nov 17, 2016
Insert Data to Tables	<ul style="list-style-type: none"> • Victor Logan 	Nov 19, 2016
Review Database & Tables	<ul style="list-style-type: none"> • Mohammed Alamoudi • Hemnthy Nalamothu • Mohammed Alduniawi 	Nov 18, 2016
Review after Inserting Data to Tables	<ul style="list-style-type: none"> • Mohammed Alamoudi • Hemnthy Nalamothu 	Nov 20,2016
Create Queries	<ul style="list-style-type: none"> • Victor Logan 	TBA
Login page	<ul style="list-style-type: none"> • Mohammed Alamoudi 	Nov/20/2016
Doctor interface	<ul style="list-style-type: none"> • Hemanth Nalamothu 	Nov/30/2016
Patient interface	<ul style="list-style-type: none"> • Mohammed Alduniawi 	Nov 30, 2016
Receptionist interface	<ul style="list-style-type: none"> • Mohammed Alamoudi 	Nov/30/2016
Admin interface	<ul style="list-style-type: none"> • Hemanth Nalamothu 	Nov/30/2016
Home Page and Design (HTML & CSS)	<ul style="list-style-type: none"> • Mohammed Alamoudi • Hemanth Nalamothu • Mohammed Alduniawi • Victor Logan 	Nov/30/2016
PHP (Database Connection & Queries)	<ul style="list-style-type: none"> • Mohammed Alduniawi 	TBA
Testing	<ul style="list-style-type: none"> • Victor Logan 	TBA

Members ranking:

Member 1 : Mohammed Alamoudi

My teammates and I agrees I did 30% of the overall project. My specific tasks include

- Task 1 : I helped Hemanth to do the ERD.
- Task 2 : I helped Hemanth to do the relational model.
- Task 3 : I coded the login/logout pages.
- Task 4 : I coded the header design.
- Task 5 : I helped writing the first, second, and final reports.
- Task 6 : I wrote the code for the receptionist.
- Task 7 : I helped doing the website design.

Member 2 : Hemanth Nalamothu

My teammates and I agrees I did 30% of the overall project. My specific tasks include

- Task 1 : I did ERD and Relational Schema.
- Task 2 : I wrote code for Doctor and Admin Pages
- Task 3 : I designed the database format and Database Requirements.
- Task 4 : I involved in completion of first, second, and final reports.

Member 3: Mohammed Alduniawi

My teammates and I agrees I did 25 % of the overall project. My specific tasks include:

- Task 1: I worked with Hemanth and Mohammed in preparing the ERD for the project.
- Task 2: I reviewed the relational schema
- Task 3: I worked in the database development process for the project
- Task 4: I worked in the patient interface and I wrote the code for the Patient pages
- Task 5: I helped in writing the first, second, and the final reports

Member 4: Victor Logan

My teammates and I agrees I did 15 % of the overall project. My specific tasks include:

- Task 1:I helped create the overview of database and wrote the first report
- Task 2: review the relational schema and ER diagram
- Task 3: insert data into database with SQL
- Task 4: I helped write the final report

Finally we uploaded the project to this link:

<http://www.malamodi.com/internetApp/login/index.php>

And those are the user id's and passwords for the four type of users to test it:

	Patient	Receptionist	Admin	Doctor
User ID	198024	12007	12003	12001
Password	198024	12007	12003	12001