

Project Cover Page

Course : INFO 7210

Semester : SPRING 2013

Instructor : Dimitrios Fotiadis

Students : Albert Carter
Di Huang
Zihang Pu
Cong Wu

Description : Homework Answers

Date Assigned : February 15, 2013

Date Due : April 15, 2013

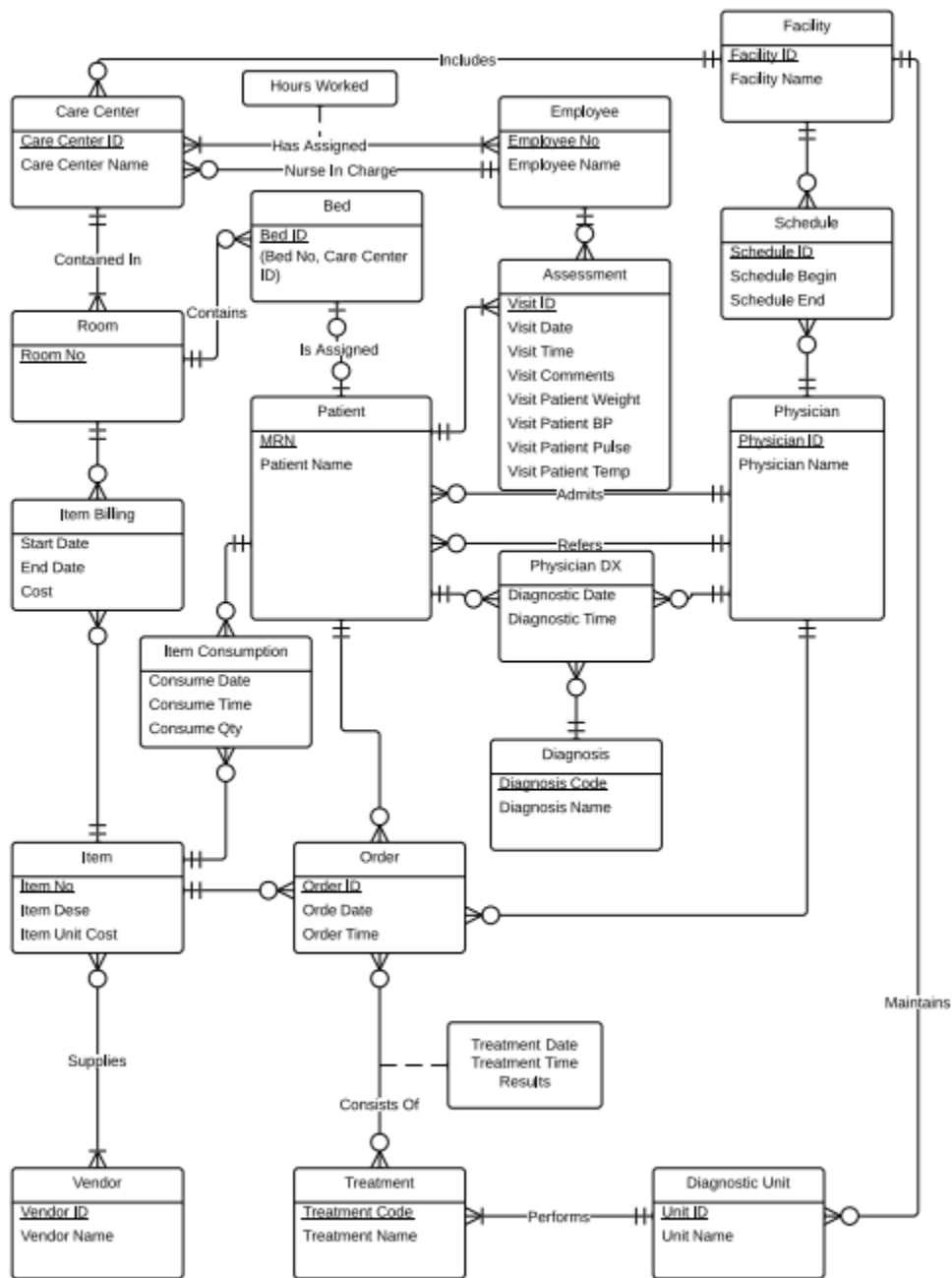
Date Submitted : April 15, 2013

Phase 1 Deliverables

- ER Diagram
- ERR Diagram
- Business Rules

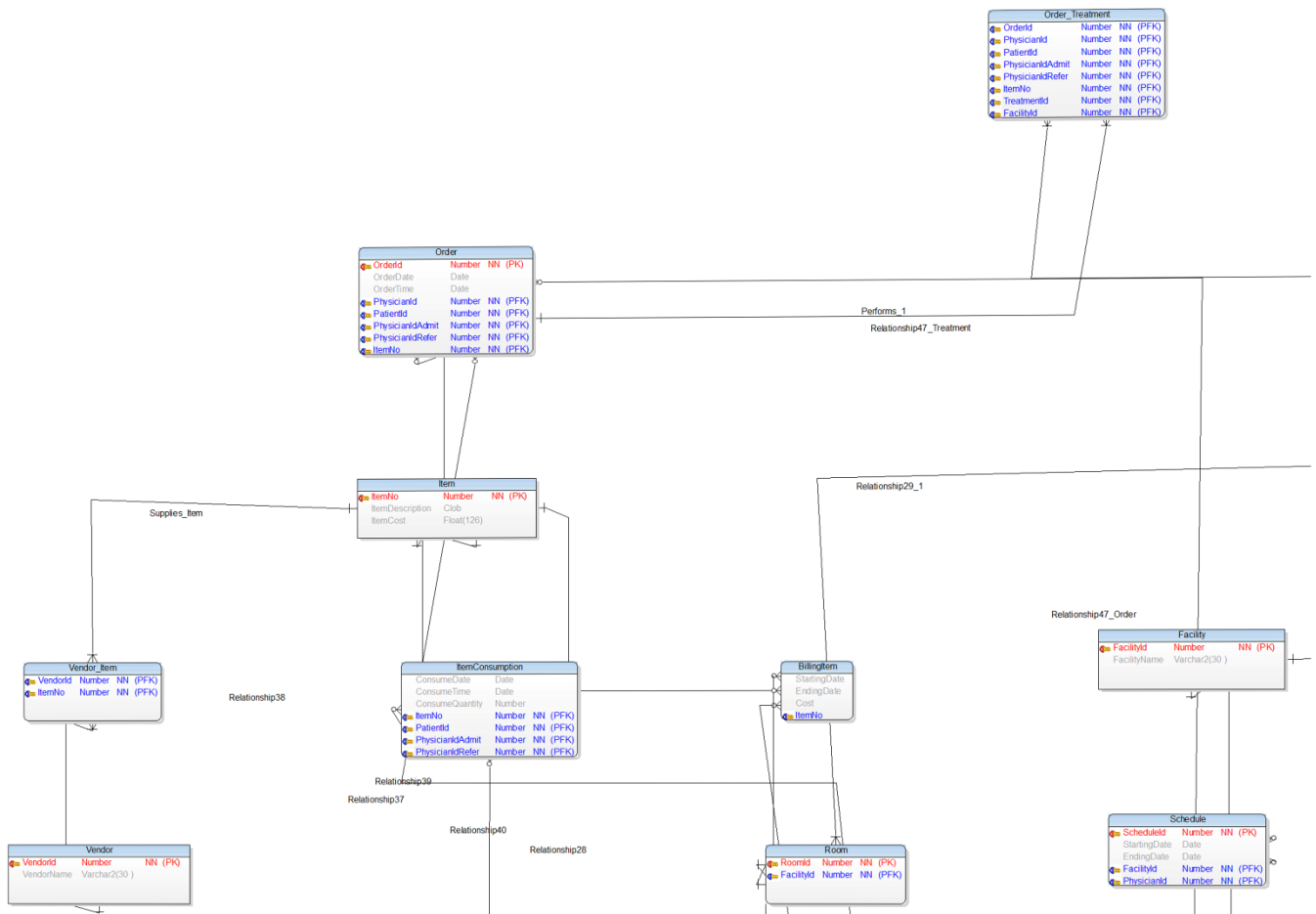
Phase I deliverables

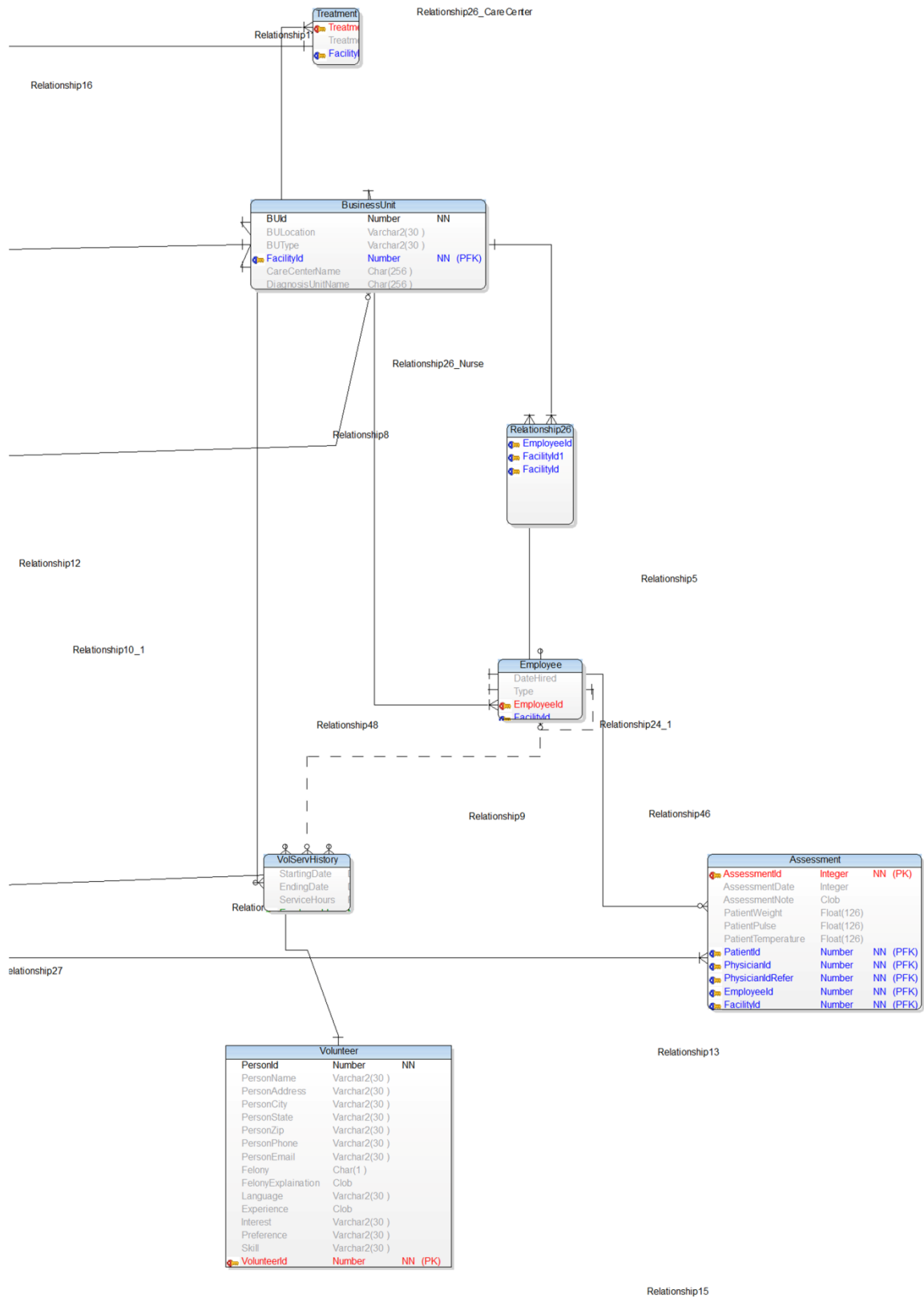
1 Develop an E-R diagram for MVCH

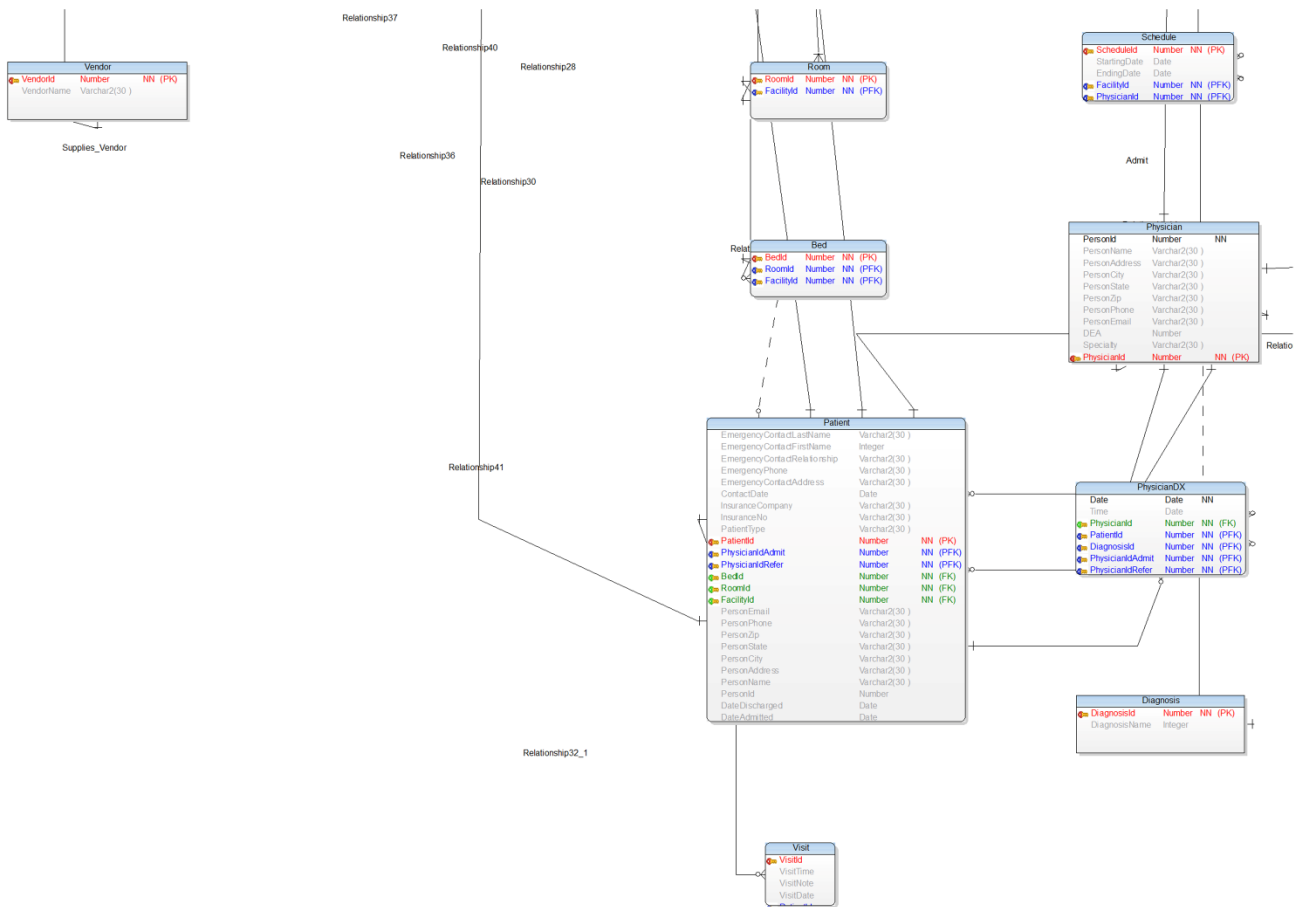


2 Draw an EER diagram of MVCH









3 Develop a list of well-stated business rules for your E-R diagram.

A DIAGNOSTIC UNIT performs one or many TREATMENTS.
A TREATMENT is performed by only one DIAGNOSTIC UNIT.
A FACILITY can have zero, one or many CARE CENTERS.
A CARE CENTER is part of one and only one FACILITY.
A FACILITY can have zero, one or many DIAGNOSTIC UNITS.
A DIAGNOSTIC UNIT is part of only one FACILITY.
A CARE CENTER has one or many EMPLOYEES.
A CARE CENTER has one EMPLOYEE assigned as morning nurse in charge.
A CARE CENTER has one EMPLOYEE assigned as nighttime nurse in charge.
A CARE CENTER contains one or many ROOMS.
A ROOM is inside only one CARE CENTER.
A ROOM has zero, one, or many BEDS.
A BED is inside only one ROOM.
A BED is assigned to zero or one RESIDENTS.
A PATIENT is either an OUTPATIENT or a RESIDENT, but is an OUTPATIENT by default.
A RESIDENT is assigned to zero or one BED.
A PATIENT has zero, one, or more VISITS.
A VISIT has one and only one PATIENT.
A FACILITY has one or many SCHEDULES for its PHYSICIANS.
Each SCHEDULE is one and only one FACILITY and for one and only one PHYSICIAN.
A PHYSICIAN can have zero, one, or many SCHEDULES.
A PHYSICIAN admits zero, one, or many PATIENTS.
A PATIENT is admitted by only one PHYSICIAN.
A PHYSICIAN refers zero, one, or many PATIENTS.
A PATIENT is referred by zero or one one PHYSICIAN.
(refers and admits are different relationships)
A PATIENT uses zero, one, or many ITEMS.
An ITEM is used by zero, one, or many PATIENTS.
An ITEM is purchased from one or many VENDORS.
A VENDOR may sell zero, one, or many ITEMS.
A PHYSICIAN gives one or more DIAGNOSES for one or more PATIENTS..
A DIAGNOSIS is given to one PATIENT by one PHYSICIAN.
A PHYSICIAN writes zero, one, or many ORDERS for one PATIENT.
An ORDER is written by one PHYSICIAN.
An ORDER includes zero, one, or more ITEMS.
An ITEM is part of zero, one, or more ORDERS.
An ORDER consists of zero, one, more TREATMENTS.
A TREATMENT is part of zero, one, or more ORDERS.
An EMPLOYEE completes zero, one, or more ASSESSMENTS of a PATIENT.
A PATIENT may have zero, one, or more ASSESSMENTS over time.
A VOLUNTEER has one emergency contact.
A VOLUNTEER has exactly two REFERENCES. A REFERENCE provides reference for one or more VOLUNTEERS.
A VOLUNTEER has exactly one most recent EMPLOYER.
An EMPLOYER employs zero, one, or many VOLUNTEERS.
A VOLUNTEER has zero, one, or more previous VOLUNTEER_EXPERIENCES.
A VOLUNTEER_EXPERIENCE is experienced by one or more VOLUNTEERS.
A VOLUNTEER has zero, one, or more SKILLS.

A SKILL is learned by zero, one, or more VOLUNTEERs.
A VOLUNTEER has a hobby. (denormalized)
A VOLUNTEER has an interest.
A INTEREST is enjoyed by one or more VOLUNTEERs.
A VOLUNTEER speaks one or more LANGUAGEs.
A LANGUAGE is spoken by zero, one, or more VOLUNTEERs.
A VOLUNTEER prefers one or more TIMESLOTs.
A TIMESLOT is chosen by zero, one, or more VOLUNTEERs.
A PATIENT has zero, one, or more INSURANCE_COMPANY.
An INSURANCE_COMPANY has one or more PATIENTs.
A VOLUNTEER is supervised by one EMPLOYEE or PHYSICIAN at a time, which is stored in VOLUNTEER_SERV_HISTORY.
A NURSE is either a REGISTERED_NURS (RN), or PRACTICAL_NURSE (LPN).
A NURSE may also be a FLOATER_NURSE (FN).
A FLOATER_NURSE cannot be assigned to a CARE_CENTER.
A RN directs zero, one, or many LPN.
A LPN is directed by only one RN.
An EMPLOYEE can be a NURSE, STAFF, and/or TECHNICIAN.
STAFF are assigned to one and only one WORK_UNIT.
TECHNICIANs are assigned to one and only one WORK_UNIT.
A WORK_UNIT has one or many STAFF.
A WORK_UNIT has one or many TECHNICIANs.

Phase 2 Deliverables

- Detailed Relational Schema
- Analyze/Diagram Functional Dependencies

Phase II Deliverables

- 1 Develop a detailed relational schema for the Mountain View Community Hospital case

DIAGNOSTIC_UNIT(DUnitName)

TREATMENT(TrtID, TrtName, *DUnitName*)

INSURANCE_COMPANY (InsuranceNo, InsName, InsAddress, InsPhone)

FACILITY (FacID, FacName)

PERSON(PID, PName, PDOB, PEmail, PAdd1, PAdd2, PCity, PState, PZip, PHomePhone, PWorkPhone, IsPhysician, IsPatient, IsEmployee, IsVolunteer)

PHYSICIAN (PhysicianID, CellNo, PagerNo, DEANo, Specialty)

PATIENT (PatientID, *AdmitPhysician*, *ReferPhysician*, *InsuranceNo*, *EmergencyPersonID*,

EmergencyPersonRelation, InsFName, InsLName, OutInPatient,

RESIDENT (RPatientID, *BedNo*, *RoomNo*, AdmitDate, DischargeDate)

OUTPATIENT(OPatientID)

NURSE (NurseID, CertDegree, StateLicenseNo, NurseSpeciality, NurseType)

RN (RNID)

LN (LNID, Supervisor)

FIELD_CERTIFICATION (FCID, NurseID, FCDescription)

CARE_CENTER (CCID, CCName, *DayInCharge*, *NightInCharge*)

ROOM (RoomNo, *CCID*)

BED (BedNo, *RoomNo*)

CARE_Center_ASSIGNMENT (CCAID, CCStart, CCEnd, HrsWorked, *CCID*, *NurseID*)

DIAGNOSIS(DiagnosisID, DiagnosisName)

PATIENT_DIAGNOSIS(PDiagnosisID, *PhysicianID*, *PatientID*, *DiagnosisCode*, DiagnosisTime)

ASSESSMENT (AssID, AssDate, Asstime, AssComment, PatientWeight, PatientBP, PatientPulse, PatientTemperature, *PatientID*, *NurseID*)

ITEM (ItemNo, ItemDesc, ItemCost)

VENDOR (VendorID, VendorName)

INVENTORY (InvID, *ItemNo*, *VendorID*)

ITEM_BILL(IBID, *ItemNo*, *RoomNo*, *PatientID*, StartDate, EndDate, Cost)

ITEM_CONSUMPTION(ICID, *ItemNo*, *PatientID*, ConsumeDate, ConsumeTime, ConsumeQty)

ORDER(OrderID, OrderTime, *PatientID*, *PhysicianID*, *ItemNo*)

TREATMENT_ORDER(TOID, TrtDate, TrtTime, Results, *OrderID*, *TrtID*)

VISIT (VisitID, VisitTime, Description, *OPatientid*)

SCHEDULE (ScheduleID, SchStart, SchEnd, *PhysicianID*, *FacID*)

WORK_UNIT(WUnitID, *FacilityID*, Floor, UnitType)

EMPLOYEE (emplID, HireDate, EmpType)

STAFF(*StaffID*, JobClass, *WUnitID*)

TECHNICIAN(TechID, *WUnitID*)

TECHNICIAN_SKILL (TSID, *TechID*, TSDescription)

VOLUNTEER (VID, *VEmergencyPersonID*, VEmergencyRelation, HadFelony, FelonyExplain, HadVolunteeExperience, WhyVolunteering, VEmpty, VEmployAdd1, VEmployAdd2, VEmployerPosition, VEmployStartDate, VEmployEndDate)

VOL_SKILL (VSID, *VID*, VSDescription)

VOL_INTEREST (VIID, *VID*, VIDescription)

VOL_LANGUAGE (VLID, *VID*, VLDescription)

VOL_EXPERIENCE (VEID, *VID*, VEDescription)

VOL_HOBBY (VHID, *VID*, VHDescription)

VOL_SERVICE (VSID, *VID*, ServiceDescription)

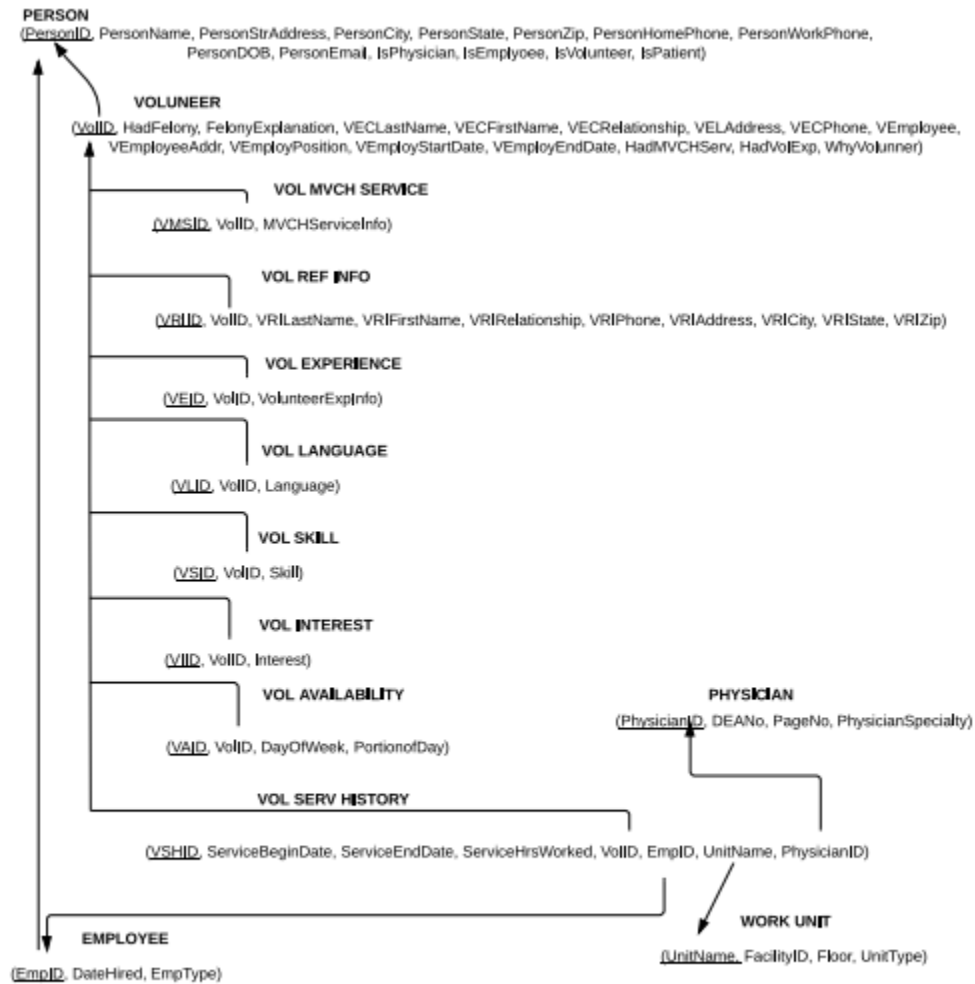
VOL_SERVICE_HISOTRY (VGSID, *WUID*, *VID*, *PhysicianID*, *EmpID*, ServiceStartDate, (VGSID, WUID, PhysicianID, EmpID) -> ServiceStartDate

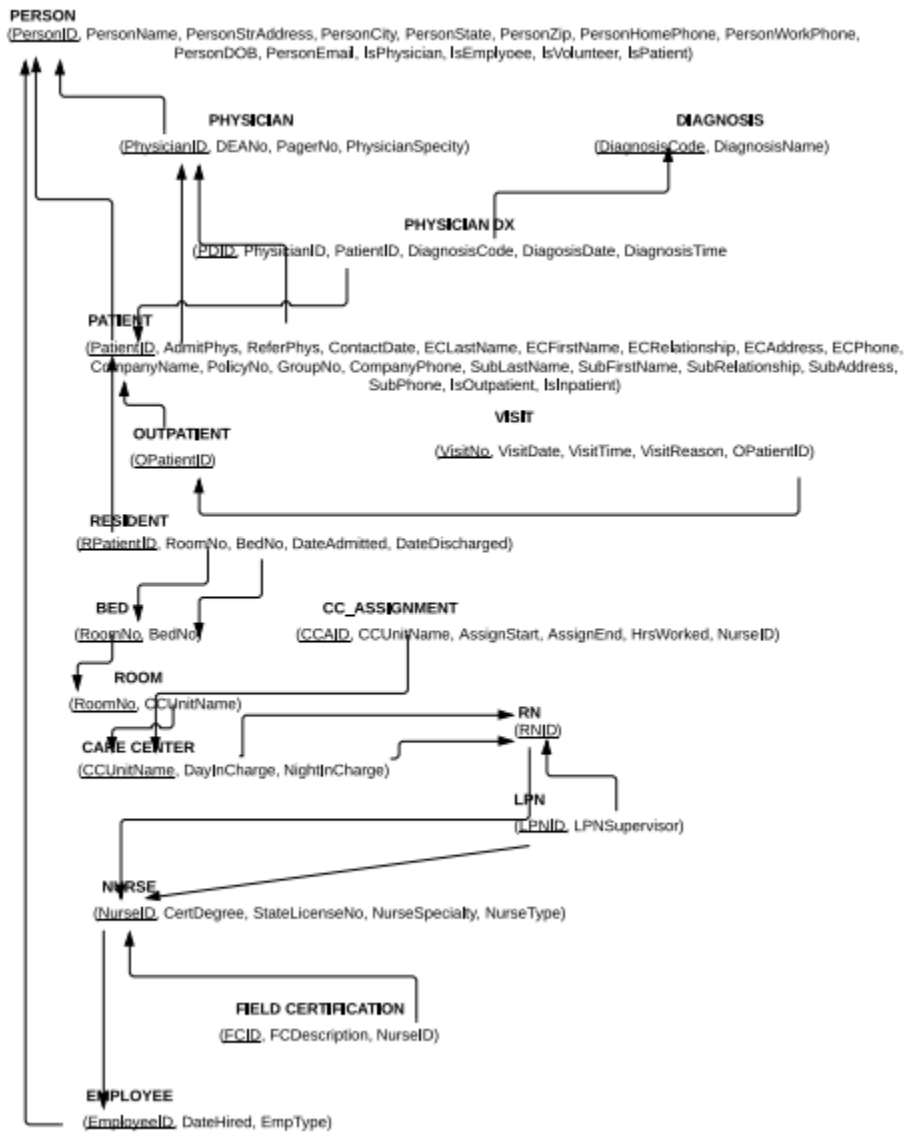
ServiceEndDate, ServiceHrsWrkd)

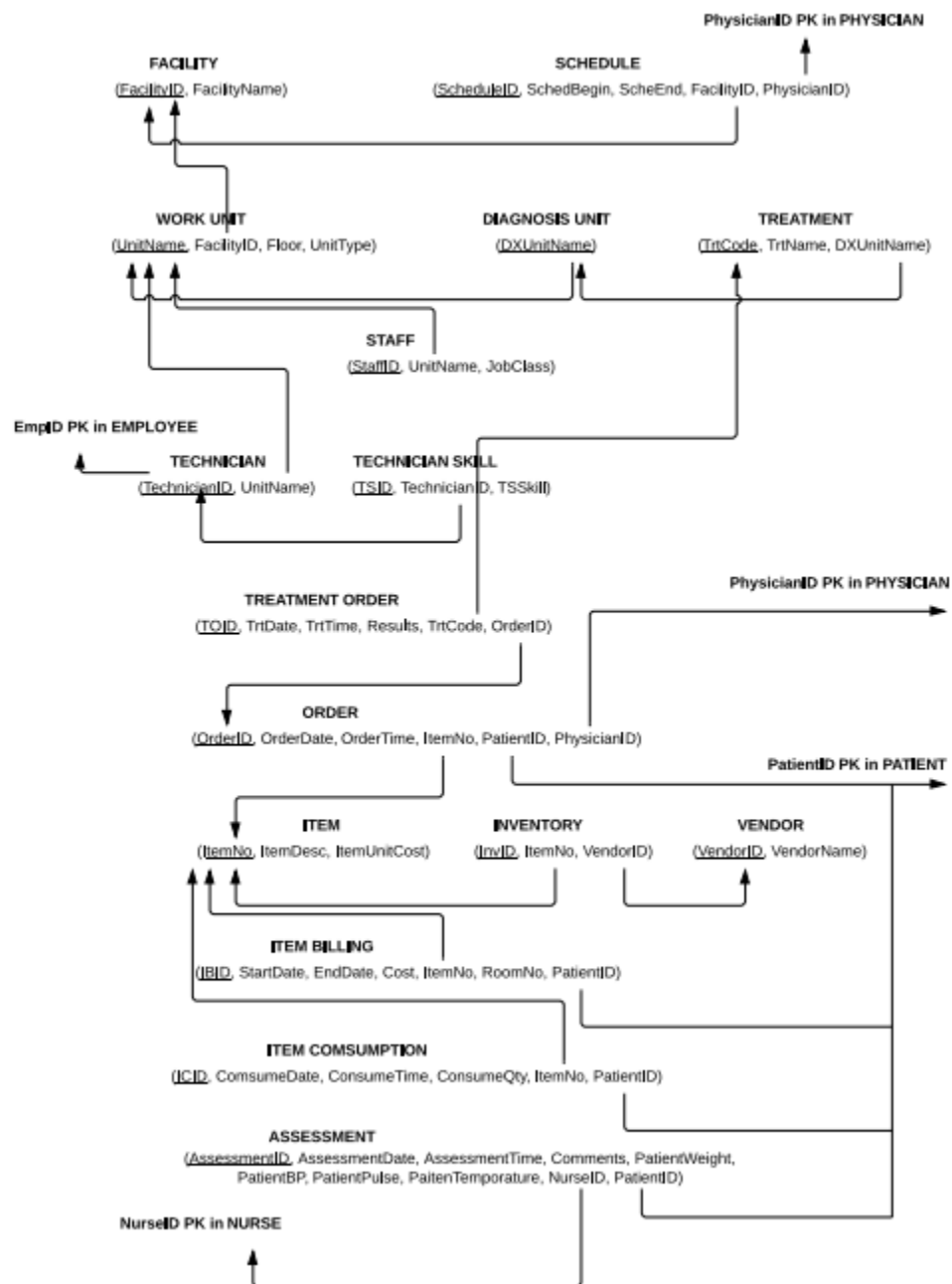
VOL_REFERENCE_INFO (VRIID, *VolIID*, *PersonID*, VRRelationship)

VOL_AVAIABILITY (VAID, Day, Shift, VolIID)

2 Analyze and diagram the functional dependencies in each relation.







Phase 3 Deliverables

- SQL Code Scripts
- .bat file
- SQL Loader

Phase III deliverables

- 1 Develop SQL code scripts that would create the database files corresponding to your relational schema for the Mountain View Community Hospital case study created in phase 2 of the class project. Organize the scripts so that you can execute them from one central command file(.bat)

```
--  
DROP TABLE ItemConsumption CASCADE CONSTRAINTS;  
DROP TABLE Relationship26 CASCADE CONSTRAINTS;  
DROP TABLE Vendor_Item CASCADE CONSTRAINTS;  
DROP TABLE Order_Treatment CASCADE CONSTRAINTS;  
DROP TABLE Visit CASCADE CONSTRAINTS;  
DROP TABLE Schedule CASCADE CONSTRAINTS;  
DROP TABLE PhysicianDX CASCADE CONSTRAINTS;  
DROP TABLE Diagnosis CASCADE CONSTRAINTS;  
DROP TABLE OOrder CASCADE CONSTRAINTS;  
DROP TABLE Assessment CASCADE CONSTRAINTS;  
DROP TABLE Treatment CASCADE CONSTRAINTS;  
DROP TABLE BillingItem CASCADE CONSTRAINTS;  
DROP TABLE VolServHistory CASCADE CONSTRAINTS;  
DROP TABLE Patient CASCADE CONSTRAINTS;  
DROP TABLE Physician CASCADE CONSTRAINTS;  
DROP TABLE Employee CASCADE CONSTRAINTS;  
DROP TABLE Volunteer CASCADE CONSTRAINTS;  
DROP TABLE Item CASCADE CONSTRAINTS;  
DROP TABLE Vendor CASCADE CONSTRAINTS;  
DROP TABLE Bed CASCADE CONSTRAINTS;  
DROP TABLE Room CASCADE CONSTRAINTS;  
DROP TABLE BusinessUnit CASCADE CONSTRAINTS;  
DROP TABLE Facility CASCADE CONSTRAINTS;  
  
CREATE TABLE Patient(  
    EmergencyContactLastName Varchar2(30 ),  
    EmergencyContactFirstName Integer,  
    EmergencyContactRelationship Varchar2(30 ),  
    EmergencyPhone Varchar2(30 ),  
    EmergencyContactAddress Varchar2(30 ),  
    ContactDate Date,  
    InsuranceCompany Varchar2(30 ),  
    InsuranceNo Varchar2(30 ),  
    PatientType Varchar2(30 ) CONSTRAINT ValidValuesPatientType CHECK ((PatientType IN  
( 'Outpatient','Resident','Discharged'))),  
    PatientId Number NOT NULL,  
    PhysicianIdAdmit Number NOT NULL,  
    PhysicianIdRefer Number NOT NULL,  
    BedId Number,  
    RoomId Number,  
    FacilityId Number,  
    PersonEmail Varchar2(30 ),
```

```

PersonPhone Varchar2(30 ),
PersonZip Varchar2(30 ),
PersonState Varchar2(30 ),
PersonCity Varchar2(30 ),
PersonAddress Varchar2(30 ),
PersonName Varchar2(30 ),
PersonId Number,
DateDischarged Date,
DateAdmitted Date
);

```

```

ALTER TABLE Patient ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY
(PatientId,PhysicianIdAdmit,PhysicianIdRefer);

```

```

CREATE TABLE Physician(
PersonId Number NOT NULL,
PersonName Varchar2(30 ),
PersonAddress Varchar2(30 ),
PersonCity Varchar2(30 ),
PersonState Varchar2(30 ),
PersonZip Varchar2(30 ),
PersonPhone Varchar2(30 ),
PersonEmail Varchar2(30 ),
DEA Number,
Specialty Varchar2(30 ),
PhysicianId Number NOT NULL
);

```

```

ALTER TABLE Physician ADD CONSTRAINT Unique_Identifier3 PRIMARY KEY (PhysicianId);

```

```

CREATE TABLE Employee(
DateHired Date,
Type Varchar2(30 ) CONSTRAINT ValidValuesType CHECK ((Type IN ('NURSE','STAFF'))),
EmployeeId Number NOT NULL,
FacilityId Number NOT NULL,
FK_EmployeeId Number NOT NULL,
FK_FacilityId Number NOT NULL,
PersonEmail Varchar2(30 ),
PersonPhone Varchar2(30 ),
PersonZip Varchar2(30 ),
PersonState Varchar2(30 ),
PersonCity Varchar2(30 ),
PersonAddress Varchar2(30 ),
PersonName Varchar2(30 ),
PersonId Number,
NurseType Varchar2(30 )
CONSTRAINT ValidValuesNurseType CHECK ((NurseType IN ('REGISTERED','LICENSED'))),
LicenseNo Number,
Degree Varchar2(30 )
);

```

);

ALTER TABLE Employee ADD CONSTRAINT Unique_Identifier4 PRIMARY KEY (EmployeeId, FacilityId);

```
CREATE TABLE Visit(  
    VisitId Number NOT NULL,  
    VisitTime Date,  
    VisitNote Clob,  
    VisitDate Date,  
    PatientId Number NOT NULL,  
    PhysicianIdAdmit Number NOT NULL,  
    PhysicianIdRefer Number NOT NULL  
);
```

ALTER TABLE Visit ADD CONSTRAINT Unique_Identifier10 PRIMARY KEY
(VisitId, PatientId, PhysicianIdAdmit, PhysicianIdRefer);

```
CREATE TABLE Bed(  
    BedId Number NOT NULL,  
    RoomId Number NOT NULL,  
    FacilityId Number NOT NULL  
);
```

ALTER TABLE Bed ADD CONSTRAINT Unique_Identifier11 PRIMARY KEY (BedId, RoomId, FacilityId);

```
CREATE TABLE Room(  
    RoomId Number NOT NULL,  
    FacilityId Number NOT NULL  
);
```

ALTER TABLE Room ADD CONSTRAINT Unique_Identifier12 PRIMARY KEY (RoomId, FacilityId);

```
CREATE TABLE Facility(  
    FacilityId Number NOT NULL,  
    FacilityName Varchar2(30 )  
);
```

ALTER TABLE Facility ADD CONSTRAINT Unique_Identifier14 PRIMARY KEY (FacilityId);

```
CREATE TABLE Schedule(  
    ScheduleId Number NOT NULL,  
    StartingDate Date,  
    EndingDate Date,  
    FacilityId Number NOT NULL,  
    PhysicianId Number NOT NULL  
);
```

ALTER TABLE Schedule ADD CONSTRAINT Unique_Identifier15 PRIMARY KEY
(ScheduleId, FacilityId, PhysicianId);

```
CREATE TABLE VolServHistory(  
    StartingDate Date,  
    EndingDate Date,  
    ServiceHours Float(126),  
    EmployeeId Number,  
    PhysicianId Number NOT NULL,  
    FacilityId Number,  
    FacilityId1 Number NOT NULL,  
    VolunteerId Number NOT NULL  
);
```

```
ALTER TABLE VolServHistory ADD CONSTRAINT Unique_Identifier17 PRIMARY KEY  
(PhysicianId, FacilityId1, VolunteerId);
```

```
CREATE TABLE BusinessUnit(  
    BUId Number NOT NULL,  
    BULocation Varchar2(30 ),  
    BUType Varchar2(30 )  
    CONSTRAINT ValidValuesBUType CHECK ((BUType IN ('Facility','Floor'))),  
    FacilityId Number NOT NULL,  
    CareCenterName Char(256 ),  
    DiagnosisUnitName Char(256 )  
);
```

```
ALTER TABLE BusinessUnit ADD CONSTRAINT Unique_Identifier19 PRIMARY KEY (FacilityId);
```

```
CREATE TABLE PhysicianDX(  
    PDate Date NOT NULL,  
    PTime Date,  
    PhysicianId Number NOT NULL,  
    PatientId Number NOT NULL,  
    DiagnosisId Number NOT NULL,  
    PhysicianIdAdmit Number NOT NULL,  
    PhysicianIdRefer Number NOT NULL  
);
```

```
ALTER TABLE PhysicianDX ADD CONSTRAINT Unique_Identifier20 PRIMARY KEY  
(PatientId, DiagnosisId, PhysicianIdAdmit, PhysicianIdRefer);
```

```
CREATE TABLE Diagnosis(  
    DiagnosisId Number NOT NULL,  
    DiagnosisName Integer  
);
```

```
ALTER TABLE Diagnosis ADD CONSTRAINT Unique_Identifier21 PRIMARY KEY (DiagnosisId);
```

```
CREATE TABLE OOrder(  
    OrderId Number NOT NULL,
```

```

OrderDate Date,
OrderTime Date,
PhysicianId Number NOT NULL,
PatientId Number NOT NULL,
PhysicianIdAdmit Number NOT NULL,
PhysicianIdRefer Number NOT NULL,
ItemNo Number NOT NULL
);

```

```

ALTER TABLE OOrder ADD CONSTRAINT Unique_Identifier22 PRIMARY KEY
(OrderId,PhysicianId,PatientId,PhysicianIdAdmit,PhysicianIdRefer,ItemNo);

```

```

CREATE TABLE Assessment(
  AssessmentId Number NOT NULL,
  AssessmentDate Integer,
  AssessmentNote Clob,
  PatientWeight Float(126),
  PatientPulse Float(126),
  PatientTemperature Float(126),
  PatientId Number NOT NULL,
  PhysicianId Number NOT NULL,
  PhysicianIdRefer Number NOT NULL,
  EmployeeId Number NOT NULL,
  FacilityId Number NOT NULL
)
/

```

```

-- Add keys for table Assessment

```

```

ALTER TABLE Assessment ADD CONSTRAINT Unique_Identifier28 PRIMARY KEY
(AssessmentId,PatientId,PhysicianId,PhysicianIdRefer,EmployeeId,FacilityId)
/

```

```

-- Table Treatment

```

```

CREATE TABLE Treatment(
  TreatmentId Number NOT NULL,
  TreatmentName Varchar2(30 ),
  FacilityId Number NOT NULL
)
/

```

```

-- Add keys for table Treatment

```

```

ALTER TABLE Treatment ADD CONSTRAINT Unique_Identifier29 PRIMARY KEY (TreatmentId,FacilityId)
/

```

```

-- Table BillingItem

```

```

CREATE TABLE BillingItem(
  StartingDate Date,
  EndingDate Date,
  Cost Float(126),
  ItemNo Number NOT NULL,
  RoomId Number NOT NULL,
  FacilityId Number NOT NULL,
  PatientId Number NOT NULL,
  PhysicianIdAdmit Number NOT NULL,
  PhysicianIdRefer Number NOT NULL
)
/

```

-- Add keys for table BillingItem

```

ALTER TABLE BillingItem ADD CONSTRAINT Unique_Identifier30 PRIMARY KEY
(ItemNo,RoomId,FacilityId,PatientId,PhysicianIdAdmit,PhysicianIdRefer)
/

```

-- Table Item

```

CREATE TABLE Item(
  ItemNo Number NOT NULL,
  ItemDescription Clob,
  ItemCost Float(126)
)
/

```

-- Add keys for table Item

```

ALTER TABLE Item ADD CONSTRAINT Unique_Identifier31 PRIMARY KEY (ItemNo)
/

```

-- Table Vendor

```

CREATE TABLE Vendor(
  VendorId Number NOT NULL,
  VendorName Varchar2(30 )
);

```

```

ALTER TABLE Vendor ADD CONSTRAINT Unique_Identifier32 PRIMARY KEY (VendorId);

```

```

DROP TABLE ItemConsumption CASCADE CONSTRAINTS;

```

```

CREATE TABLE ItemConsumption(
  ConsumeDate Date,
  ConsumeTime Date,
  ConsumeQuantity Number,
  ItemNo Number NOT NULL,
  PatientId Number NOT NULL,

```

```

PhysicianIdAdmit Number NOT NULL,
PhysicianIdRefer Number NOT NULL,
CONSTRAINT Unique_Identifier33 PRIMARY KEY (ItemNo, PatientId, PhysicianIdAdmit, PhysicianIdRefer)
);

```

```

CREATE TABLE Volunteer(
  PersonId Number NOT NULL,
  PersonName Varchar2(30 ),
  PersonAddress Varchar2(30 ),
  PersonCity Varchar2(30 ),
  PersonState Varchar2(30 ),
  PersonZip Varchar2(30 ),
  PersonPhone Varchar2(30 ),
  PersonEmail Varchar2(30 ),
  Felony Char(1 ),
  FelonyExplanation Clob,
  VLanguage Varchar2(30 ),
  Experience Clob,
  Interest Varchar2(30 ),
  Preference Varchar2(30 ),
  Skill Varchar2(30 ),
  VolunteerId Number NOT NULL
)
/

```

-- Add keys for table Volunteer

```

ALTER TABLE Volunteer ADD CONSTRAINT Unique_Identifier7 PRIMARY KEY (VolunteerId)
/

```

-- Table Relationship26

```

CREATE TABLE Relationship26(
  EmployeeId Number NOT NULL,
  FacilityId1 Number NOT NULL,
  FacilityId Number NOT NULL
)
/

```

-- Add keys for table Relationship26

```

ALTER TABLE Relationship26 ADD CONSTRAINT Key1 PRIMARY KEY (EmployeeId, FacilityId1, FacilityId)
/

```

-- Table Vendor_Item

```

CREATE TABLE Vendor_Item(
  VendorId Number NOT NULL,
  ItemNo Number NOT NULL

```

)
/

-- Table Order_Treatment

```
CREATE TABLE Order_Treatment(  
  OrderId Number NOT NULL,  
  PhysicianId Number NOT NULL,  
  PatientId Number NOT NULL,  
  PhysicianIdAdmit Number NOT NULL,  
  PhysicianIdRefer Number NOT NULL,  
  ItemNo Number NOT NULL,  
  TreatmentId Number NOT NULL,  
  FacilityId Number NOT NULL  
)  
/
```

-- Create relationships section -----

```
ALTER TABLE Schedule ADD CONSTRAINT Relationship5 FOREIGN KEY (FacilityId) REFERENCES  
Facility (FacilityId)  
/
```

```
ALTER TABLE VolServHistory ADD CONSTRAINT Relationship8 FOREIGN KEY (EmployeeId, FacilityId)  
REFERENCES Employee (EmployeeId, FacilityId)  
/
```

```
ALTER TABLE VolServHistory ADD CONSTRAINT Relationship9 FOREIGN KEY (PhysicianId)  
REFERENCES Physician (PhysicianId)  
/
```

```
ALTER TABLE Employee ADD CONSTRAINT Relationship10_1 FOREIGN KEY (FacilityId) REFERENCES  
BusinessUnit (FacilityId)  
/
```

```
ALTER TABLE BusinessUnit ADD CONSTRAINT Relationship11 FOREIGN KEY (FacilityId) REFERENCES  
Facility (FacilityId)  
/
```

```
ALTER TABLE VolServHistory ADD CONSTRAINT Relationship12 FOREIGN KEY (FacilityId1)  
REFERENCES BusinessUnit (FacilityId)  
/
```

```
ALTER TABLE PhysicianDX ADD CONSTRAINT Relationship13 FOREIGN KEY (PhysicianId) REFERENCES  
Physician (PhysicianId)  
/
```

```
ALTER TABLE PhysicianDX ADD CONSTRAINT Relationship14 FOREIGN KEY (PatientId, PhysicianIdAdmit,  
PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
```


/

```
ALTER TABLE PhysicianDX ADD CONSTRAINT Relationship15 FOREIGN KEY (DiagnosisId) REFERENCES  
Diagnosis (DiagnosisId)
```

/

```
ALTER TABLE OOrder ADD CONSTRAINT Relationship16 FOREIGN KEY (PhysicianId) REFERENCES  
Physician (PhysicianId)
```

/

```
ALTER TABLE Patient ADD CONSTRAINT Admit FOREIGN KEY (PhysicianIdAdmit) REFERENCES  
Physician (PhysicianId)
```

/

```
ALTER TABLE Patient ADD CONSTRAINT Refer FOREIGN KEY (PhysicianIdRefer) REFERENCES  
Physician (PhysicianId)
```

/

```
ALTER TABLE Employee ADD CONSTRAINT Relationship24_1 FOREIGN KEY (FK_EmployeeId,  
FK_FacilityId) REFERENCES Employee (EmployeeId, FacilityId)
```

/

```
ALTER TABLE Relationship26 ADD CONSTRAINT Relationship26_Nurse FOREIGN KEY (EmployeeId,  
FacilityId1) REFERENCES Employee (EmployeeId, FacilityId)
```

/

```
ALTER TABLE Relationship26 ADD CONSTRAINT Relationship26_CareCenter FOREIGN KEY (FacilityId)  
REFERENCES BusinessUnit (FacilityId)
```

/

```
ALTER TABLE Assessment ADD CONSTRAINT Relationship27 FOREIGN KEY (PatientId, PhysicianId,  
PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
```

/

```
ALTER TABLE OOrder ADD CONSTRAINT Relationship28 FOREIGN KEY (PatientId, PhysicianIdAdmit,  
PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
```

/

```
ALTER TABLE Room ADD CONSTRAINT Relationship29_1 FOREIGN KEY (FacilityId) REFERENCES  
BusinessUnit (FacilityId)
```

/

```
ALTER TABLE Bed ADD CONSTRAINT Relationship30 FOREIGN KEY (RoomId, FacilityId) REFERENCES  
Room (RoomId, FacilityId)
```

/

```
ALTER TABLE Patient ADD CONSTRAINT Relationship31_1 FOREIGN KEY (BedId, RoomId, FacilityId)  
REFERENCES Bed (BedId, RoomId, FacilityId)
```

/

ALTER TABLE Visit ADD CONSTRAINT Relationship32_1 FOREIGN KEY (PatientId, PhysicianIdAdmit, PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
/

ALTER TABLE Treatment ADD CONSTRAINT Performs_1 FOREIGN KEY (FacilityId) REFERENCES BusinessUnit (FacilityId)
/

ALTER TABLE ItemConsumption ADD CONSTRAINT Relationship36 FOREIGN KEY (ItemNo) REFERENCES Item (ItemNo)
/

ALTER TABLE BillingItem ADD CONSTRAINT Relationship37 FOREIGN KEY (ItemNo) REFERENCES Item (ItemNo)
/

ALTER TABLE BillingItem ADD CONSTRAINT Relationship38 FOREIGN KEY (RoomId, FacilityId) REFERENCES Room (RoomId, FacilityId)
/

ALTER TABLE BillingItem ADD CONSTRAINT Relationship39 FOREIGN KEY (PatientId, PhysicianIdAdmit, PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
/

ALTER TABLE OOrder ADD CONSTRAINT Relationship40 FOREIGN KEY (ItemNo) REFERENCES Item (ItemNo)
/

ALTER TABLE ItemConsumption ADD CONSTRAINT Relationship41 FOREIGN KEY (PatientId, PhysicianIdAdmit, PhysicianIdRefer) REFERENCES Patient (PatientId, PhysicianIdAdmit, PhysicianIdRefer)
/

ALTER TABLE Vendor_Item ADD CONSTRAINT Supplies_Vendor FOREIGN KEY (VendorId) REFERENCES Vendor (VendorId)
/

ALTER TABLE Vendor_Item ADD CONSTRAINT Supplies_Item FOREIGN KEY (ItemNo) REFERENCES Item (ItemNo)
/

ALTER TABLE Assessment ADD CONSTRAINT Relationship44_1 FOREIGN KEY (EmployeeId, FacilityId) REFERENCES Employee (EmployeeId, FacilityId)
/

ALTER TABLE Schedule ADD CONSTRAINT Relationship46 FOREIGN KEY (PhysicianId) REFERENCES Physician (PhysicianId)
/

```
ALTER TABLE Order_Treatment ADD CONSTRAINT Relationship47_Order FOREIGN KEY (OrderId,  
PhysicianId, PatientId, PhysicianIdAdmit, PhysicianIdRefer, ItemNo) REFERENCES OOrder (OrderId,  
PhysicianId, PatientId, PhysicianIdAdmit, PhysicianIdRefer, ItemNo)  
/
```

```
ALTER TABLE Order_Treatment ADD CONSTRAINT Relationship47_Treatment FOREIGN KEY (TreatmentId,  
FacilityId) REFERENCES Treatment (TreatmentId, FacilityId)  
/
```

```
ALTER TABLE VolServHistory ADD CONSTRAINT Relationship48 FOREIGN KEY (VolunteerId)  
REFERENCES Volunteer (VolunteerId)  
/
```

2 Populate all the tables it with sample data using SQL*Loader .

Bat file:

```
sqlldr dee/password control=physician.ctl
sqlldr dee/password control=facility.ctl
sqlldr dee/password control=schedule.ctl
sqlldr dee/password control=businessunit.ctl
sqlldr dee/password control=diagnosis.ctl
sqlldr dee/password control=treatment.ctl
sqlldr dee/password control=item.ctl
sqlldr dee/password control=vendor.ctl
sqlldr dee/password control=volunteer.ctl
sqlldr dee/password control=vendor_item.ctl
sqlldr dee/password control=room.ctl
sqlldr dee/password control=bed.ctl
sqlldr dee/password control=patient.ctl
sqlldr dee/password control=visit.ctl
sqlldr dee/password control=physiciandx.ctl
sqlldr dee/password control=oorder.ctl
sqlldr dee/password control=billingitem.ctl
sqlldr dee/password control=itemconsumption.ctl
sqlldr dee/password control=order_treatment.ctl
sqlldr dee/password control=employee.ctl
sqlldr dee/password control=volservhistory.ctl
sqlldr dee/password control=assessment.ctl
sqlldr dee/password control=relationship26.ctl
```

pause

Ctl Files:

load data

infile 'physician.csv'

into table physician

fields terminated by "," optionally enclosed by ""

("PERSONID","PERSONNAME","PERSONADDRESS","PERSONCITY","PERSONSTATE","PERSONZIP","PERSONPHONE","PERSONEMAIL","DEA","SPECIALTY","PHYSICIANID")

load data

infile 'facility.csv'

into table facility

fields terminated by "," optionally enclosed by ""

("FACILITYID","FACILITYNAME")

load data

infile 'schedule.csv'

into table schedule

fields terminated by "," optionally enclosed by ""

("SCHEDULEID","STARTINGDATE","ENDINGDATE","FACILITYID","PHYSICIANID")

```
load data
infile 'businessunit.csv'
into table businessunit
fields terminated by "," optionally enclosed by '"'
("BUID","BULOCATION","BUTYPE","FACILITYID","CARECENTERNAME","DIAGNOSISUNITNAME")
```

```
load data
infile 'diagnosis.csv'
into table diagnosis
fields terminated by "," optionally enclosed by '"'
("BUID","BULOCATION","BUTYPE","FACILITYID","CARECENTERNAME","DIAGNOSISUNITNAME")
```

```
load data
infile 'treatment.csv'
into table treatment
fields terminated by "," optionally enclosed by '"'
("TREATMENTID","TREATMENTNAME","FACILITYID")
```

```
load data
infile 'item.csv'
into table item
fields terminated by "," optionally enclosed by '"'
("ITEMNO","ITEMDESCRIPTION","ITEMCOST")
```

```
load data
infile 'vendor.csv'
into table vendor
fields terminated by "," optionally enclosed by '"'
("VENDORID","VENDORNAME")
```

```
load data
infile 'volunteer.csv'
into table volunteer
fields terminated by "," optionally enclosed by '"'
("PERSONID","PERSONNAME","PERSONADDRESS","PERSONCITY","PERSONSTATE","PERSONZIP","PERSONPHONE","PERSONEMAIL","FELONY","FELONYEXPLANATION","VLANGUAGE","EXPERIENCE","INTEREST","PREFERENCE","SKILL","VOLUNTEERID")
```

```
load data
infile 'vendor_item.csv'
into table vendor_item
fields terminated by "," optionally enclosed by '"'
("VENDORID","ITEMNO")
```

```
load data
infile 'room.csv'
into table room
fields terminated by "," optionally enclosed by '"'
```

("ROOMID","FACILITYID")

load data

infile 'bed.csv'

into table bed

fields terminated by "," optionally enclosed by ""

("BEDID","ROOMID","FACILITYID")

load data

infile 'patient.csv'

into table patient

fields terminated by "," optionally enclosed by ""

("EMERGENCYCONTACTLASTNAME","EMERGENCYCONTACTFIRSTNAME","EMERGENCYCONTACTRELATIONSHIP","EMERGENCYPHONE","EMERGENCYCONTACTADDRESS","CONTACTDATE","INSURANCECOMPANY","INSURANCENO","PATIENTTYPE","PATIENTID","PHYSICIANIDADMIT","PHYSICIANIDREFER","BEDID","ROOMID","FACILITYID","PERSONEMAIL","PERSONPHONE","PERSONZIP","PERSONSTATE","PERSONCITY","PERSONADDRESS","PERSONNAME","PERSONID","DATEDISCHARGED","DATEADMITTED")

load data

infile 'visit.csv'

into table visit

fields terminated by "," optionally enclosed by ""

("VISITID","VISITTIME","VISITNOTE","VISITDATE","PATIENTID","PHYSICIANIDADMIT","PHYSICIANIDREFER")

load data

infile 'physiciandx.csv'

into table physiciandx

fields terminated by "," optionally enclosed by ""

("PDATE","PTIME","PHYSICIANID","PATIENTID","DIAGNOSISID","PHYSICIANIDADMIT","PHYSICIANIDREFER")

load data

infile 'oorder.csv'

into table oorder

fields terminated by "," optionally enclosed by ""

("ORDERID","ORDERDATE","ORDERTIME","PHYSICIANID","PATIENTID","PHYSICIANIDADMIT","PHYSICIANIDREFER","ITEMNO")

load data

infile 'billingitem.csv'

into table billingitem

fields terminated by "," optionally enclosed by ""

("STARTINGDATE","ENDINGDATE","COST","ITEMNO","ROOMID","FACILITYID","PATIENTID","PHYSICIANIDADMIT","PHYSICIANIDREFER")

load data

infile 'itemconsumption.csv'

```

into table itemconsumption
fields terminated by "," optionally enclosed by ""
("CONSUMEDATE","CONSUMETIME","CONSUMEQUANTITY","ITEMNO","PATIENTID","PHYSICIAN
IDADMIT","PHYSICIANIDREFER")

```

```

load data
infile 'order_treatment.csv'
into table order_treatment
fields terminated by "," optionally enclosed by ""
("ORDERID","PHYSICIANID","PATIENTID","PHYSICIANIDADMIT","PHYSICIANIDREFER","ITEMNO",
"TREATMENTID","FACILITYID")

```

```

load data
infile 'employee.csv'
into table employee
fields terminated by "," optionally enclosed by ""
("DATEHIRED","TYPE","EMPLOYEEID","FACILITYID","FK_EMPLOYEEID","FK_FACILITYID","PERS
ONEMAIL","PERSONPHONE","PERSONZIP","PERSONSTATE","PERSONCITY","PERSONADDRES
S","PERSONNAME","PERSONID","NURSETYPE","LICENSENO","DEGREE")

```

```

load data
infile 'volservhistory.csv'
into table volservhistory
fields terminated by "," optionally enclosed by ""
("STARTINGDATE","ENDINGDATE","SERVICEHOURS","EMPLOYEEID","PHYSICIANID","FACILITYI
D","FACILITYID1","VOLUNTEERID")

```

```

load data
infile 'assessment.csv'
into table assessment
fields terminated by "," optionally enclosed by ""
("ASSESSMENTID","ASSESSMENTDATE","ASSESSMENTNOTE","PATIENTWEIGHT","PATIENTPU
LSE","PATIENTTEMPERATURE","PATIENTID","PHYSICIANID","PHYSICIANIDREFER","EMPLOYEEI
D","FACILITYID")

```

```

load data
infile 'relationship26.csv'
into table relationship26
fields terminated by "," optionally enclosed by ""
("EMPLOYEEID","FACILITYID1","FACILITYID")

```

```

389353.99,'Langham','21 East 74th
Street','Delhi','xxV016lw','uBrtr8',NULL,'WillBarbee@yahoo.us',NULL,'nxNRqPfdZ1XrrxwFAA7pqg56T',1
98.95,'Cramer','71 Prospect Hill',NULL,'2','yAA25SwWOJOi4','1-585-3130',NULL,86.42,'m1r',2
65673.91,'Hoogbandt','6481 Reeves Drive','Ludhiana','oEhjSn6uJGAMote7y','g8','1-409-
0257','HankMcCormick@excite.nl',457.68,NULL,3
474667.73,'Lezniak','2','Bursa','rUBbnYdMO641uxCtaNmINKNWv','M86rld','614-982-
6164','Hank.Linton1@mobileme.com',653883.88,'MpGDfmm2J',4
747108.83,'Korkovski',NULL,NULL,NULL,NULL,'819-740-8785',NULL,515.46,'WEw8Si7n7qPN',5

```

161389.89,NULL,'1 S. Riverside
Plaza','Delhi','kLKgr2HMmBOd6h6CVp','zZaaTXHd1vDoS4RQpH2Cq',NULL,'K.Moore@gawab.es',73064.42,'
YcGWYRkj7dP',6
390.02,'Mulders','6 Unbridle Way','Xian','BpxbrBI8F','Anw7TU8','478-317-
2061','BJulieze4@excite.be',616.01,'uhSp2kGfn',7
2634.51,'Ayers','5 Hunt Valley Road','Changchun','uc2NVJ','tiNm6WkltuUbz6p2It','1-517-329-
2597',NULL,4.33,'l44NvkQbDoYexQsK',8
340.08,NULL,NULL,'Shijiazhuang','SoF52NISaY','BXHhu5TsIO','1-796-783-
8591','Hank.Haynes@aol.it',77005.78,NULL,9
8489.16,'van der Laar','2280 Shallowford Rd','Vadodara','HCKeb5D','JKpwm6YXcQbusyHiswNI','1-172-933-
8434',NULL,3.97,NULL,10

1,'nNNFME05L1nJr2L'
2,'wQV5kf6Uy0jX8pezlVb7aL'
3,'m'
4,NULL
5,NULL
6,'Dw1ew6rHP'
7,'VDyO3oRLLmVx'
8,'7nH3tR6dBqCtWaJRLC67d1'
9,'YBIHpIMh2z2DqgfuBQs'
10,NULL
11,'Bs'
12,'th'
13,'L55TGpGGk'
14,'AzEyasCGsHdjyCko8G1IEcsJnO7'
15,'IMQfgHMPZRI7hth2PKS'
16,'nZMUk0M'
17,NULL
18,'7YhUc2luKU'
19,'fc5'
20,'UKlwcvQ5teJNrtIBc14nD'
21,'V8jF'
22,NULL
23,'GioMhmORrsYwluiltZRH'
24,'Z0G6jr6jGFyhgrA5QEL7pG4t5Jh0P'
25,'ZOKFxxkvQn1ED1lp'
26,'p4DtgT2YOmFDEbMQv'
27,'M'
28,'XZ5whn3iyO'
29,'WuBevmZ5uNyKmZXh6'
30,'EdZ61ImqxSrl2J2IT'
31,NULL
32,'pFGoW0Mj4sq5Uv'
33,'eQVyheGBPIXQ5f3Qho2ji'
34,'LsS2'
35,'DTmbjHHqNeuHa'
36,'OXswz3UW4qrrwuZHaw5wKNQGc'
37,'bFGoVZxiacZ16JU8h0Uix6lQ7J'

38,'hJIB5zydS7zMcf1Wm1InmED'
39,'t2OI3HP7e5Pp4EUx2Z7FH8y'
40,'TZJ8acp87UDVMYJuC2'
41,'Zz');
42,'cJKvh0IS'
43,'xeNiPzn6pLRZPsk'
44,'X0YB8FGC0fiA'
45,'iFYjd2CZeF4JksharPmG0BaZtH'
46,'ESHipwBr7BggRWYqKz2kQHBq0s7UD'
47,'7VPQqy4FCI0Z1RTUvhavKvWoDX'
48,NULL
49,'Ur1dgu06R4i4NnQzXI'
50,'EhILDZ5'

1,'21-Mar-07',NULL,2,3
2,NULL,NULL,5,6
3,'17-Dec-07','27-Feb-06',1,4
5,'08-Jul-04','20-Mar-07',4,7
6,'21-Nov-08','14-Mar-06',8,9
7,'08-Sep-04','25-Feb-01',14,8
8,NULL,'13-May-09',12,7
9,'02-Aug-03','10-Dec-07',15,4
10,'22-Apr-12',NULL,10,9

817.42,NULL,'Facility',1,'bBJTtM76gcCCpheNNLcnK6eCVQAabKCYnVGewOand88yd7saa6QHT34pWwPigIF
KDcdQVr87FjDIT1oqpbj2iioiUQoqNYsDmtl1Vvu0NxJsZjXRntUiITAoZ',NULL
541005.17,'z','Floor',4,NULL,'YCyBDbJZQGIZWFKCq6CoWUAi4GFSloKMmdzCVOWgSnfScYviPu0DYNUu
O28KVC8X7W1atsjuZBTtFz4mml5tJOfIFTcZryJfWIIlM4twoXajh14VDBAyO3gca5oMALAC6470R7pNK78FSN
ttafz0B6L6UMLyMfZ8U'
8837.8,'eCLGTDksQxr','Facility',6,'Y3qqjdZ0DE0CMAKXjyFTkiNWneM61wgbWcU2NIprimpGmOlVjir3mGDabR
0zC0Nko0YbV1AgcBmxx4NZlbitSsIlxy5usFApFvxJsKziSPg37dM1hL5MyrzXtYiMQR4qsiz02XG2NIOXoOeVu
bOM0VmUeh1hsublqMZAA1LjWW2l2Nvl27qTCreygLonW2CeWAMRBejRU0upGQ6pZYd6UGwshgllO1tB4N
CC1cXkHkOXaRaJ1myoDU','g7shaKEs5bKlq'
110857.52,'YZmZoR6qtvhKcTt','Floor',7,'1WXzn63Erv0HyeOx4EH5HVliMpFemOop1B2i1IQKVGfThSUSci0Yp
Kbq0HyARCAitmGhBzWuQwwj5Qu3YEzB3Eyl7ejZid8MPgM4wFReC6JIMcRWxnUS2kP2OQtm5kxVS','titzQg
1ps72qWCh6l2YEfl63N2tMQm'
14.52,'VeKjJpjo5L75lDesRSOWf','Floor',5,'GvZi2tOwlWmXTtnSJcOSb6HCs8lQ1o2ly8vBukn15T6Bgfj3ycazF0
LerDovsXi1pbxhPHR5oCpjVKShNIqPTIZl3KroAdND','Rh00hsukkvYgjGndJcqa1VboorZQkyvPJcqwezXgRCA
RuwcqjLWuDzu1uhYpQTryBS27vIUog7ANUK5BGZJxQ1svpCPYNgPMOZ0dTB0WmUdzBhGpPHCc8Whb18
0CcROcheNKfbSLeMbxXpudVDpNPJTziOD'
9202.52,'Nn0zbgJsZGjLh2x3lEd6dhV','Facility',11,'KQZJQeJAFZArv5fsF8DRF7OMFMYs78ubZJEIxnM5ltW3x
vRYapGiY7BMflxUmKWqUHNTmVVYF7oDRyW8fuTmFPCnitPTkq6BEBZUcTaBnuBUfzdnGS60TOnW86uKO
oVCmZEd7YHXf2YwSNQWMXfSCCGvTTA441DBJWICImuDRItLQ1vJOMZTbrD3gwlrZRXJry2iABziEqpxskN
DeWb6','EaKDIouCrtZaLZllhxRwjKwxdD8EgrugnKWsqcVt4YnmDVtU4KLrL100R4l7Vle2Z'
93.6,'nRaNW','Facility',12,NULL,'ReIQCZ5dtUwtiSPGtQMHKQl1MHT8emHjdh5hT0O1LmLydHNbHiVpX7OHw
cbpCVEeeGE'
275.78,'7KixWNmOIs','Facility',15,'H74yYTtOLm2gAmNb3ppSLPRcQgAJvjf7mgSmT0aZ2WxnemexyjllsKG2w
BpJRIctKSr4eJ7Qpfuz1SsblLpj8WbXVb32dYl6EjbqMSeYJtSwlOjQA1CwzLS78eQKuZK3pslVxswPWth0McZj
LCO5sorZOCYAi2lJkymUHEoaNSZBhisrbcXsvL6Y1NzOx0nn','iSKesFHzoTTwdxddQXkuaoZJklIHYTJFO7eP

zluVpZglrBXEfubiHEdfoB036kJrrJegcEiQJx57OXsS8FaVYCbgrlCH85lf40sLNbBtGMxbPPnnfRDC20UXFmgSI
Hy1u'
12067.27,NULL,'Facility',16,NULL,'fGuA8Mcg6AWuG1it0vHF1wOpF43ugFC0gRpHL5JQPvyxgZxomwxgivDsa
vfX33KI4EfU7N1lxO0oz1uxZbzGEhw8zFnpjallHdofZ8GumX0mHhZjV4A3oo1ibUNs0nKuiCNjAOsQzKdtvv5rA
VKTY6wFfYx6Pv1PKuLOr7K0qg5ZCwuuB3ISbdr'

1,28.73
2,84.8
3,NULL
4,7707.59
5,67.9
6,909511.39
7,6.94
8,9.74
9,NULL
10,35.31

1,'ELtP7rDTTGq',4
2,'JkT0pX7hUeE8iuUjyCc',4
3,'cW5Z2v3',15
4,NULL,11
5,'INEyjcTu0GhbNHHkunb',15
6,'EWyF',4
7,'PwmsBeL0WcUq2C6wVPaNjQbrvSMr',11
8,'B3',15
9,'dGqdhcuZuMB',1
10,'VzaFqVXeX6LZ5gDANJ',4

1,EMPTY_CLOB(),7.20
2,EMPTY_CLOB(),1370.04
3,EMPTY_CLOB(),NULL
4,EMPTY_CLOB(),NULL
5,EMPTY_CLOB(),65.08
6,EMPTY_CLOB(),NULL
7,EMPTY_CLOB(),04.31
8,EMPTY_CLOB(),52.84
9,EMPTY_CLOB(),86.57
10,EMPTY_CLOB(),7.35

1,NULL
2,NULL
3,'Endemol'
4,'Lilly'
5,'Smith Kline French'
6,'Samsung'
7,'Quarterdeck'
8,'Silicon Graphics'
9,'Neerдам Installaties'
10,NULL

988.64,'Mairy','3 Sarasota St.',NULL,'maZkJnS8rKbgav','FiRF1ASA0','167-8967','K.Herzog2@libero.no','t',EMPTY_CLOB(),'2alkmCbekB67c1STR',EMPTY_CLOB(),NULL,'U4pN',NULL,1
 13554.18,'Robertson','41 Gascony Pl.',NULL,'Oeh37xI8eaW','pqr8YzA61Lm83fjafNzGi6wz','1-702-107-9142','S.Schmidt2@hotmail.cc','V',EMPTY_CLOB(),'bmSHLfCUsUu7HMh3NTLqJEz2z',EMPTY_CLOB(),'0FcoRJBhiWVJ0rF',NULL,'vi8Fyokf',2
 300190.13,'Schlee','3 Wellington Dr.','Suizhou','0llu5FEH1jfLhyuc3Q4EaRcLczsbC',NULL,'190-8470','Ciska.Arden@kpn.nl',NULL,EMPTY_CLOB(),'U3IUyQ6ePalsyt6m01vZ',EMPTY_CLOB(),'5S3teGKXUayQS0TwCrE0zbU',NULL,'aA5CaX2cLK',3
 918151.73,'King','0 S. Brackbank Dr','Dongtai','YY6ZFuBX8nUjR','TZPkekx1jSxS4tI5UqA','1-963-425-3959',NULL,NULL,EMPTY_CLOB(),NULL,EMPTY_CLOB(),'NuCDRVQ5TGeGBeoZGit1nJXBx1Ab1','yr7bV','l0MydDWN8sTxJ7xPgvWBmRttW0xy2F',4
 70503.74,'Paddock','27 N. Deerwood Street','Kharkov',NULL,NULL,'1-327-2094','SjorsAhlgren@gawab.fr','n',EMPTY_CLOB(),'wArITvfnuLyD8roAsGikKcmZf',EMPTY_CLOB(),'peW0QZSINSAwL','Bt50wcQ3RpOyMXzghTYab7sSm','6aiG7Jb47tvCVpJdrht8kR',5
 793.84,NULL,'399 Ninth Avenue','Haicheng',NULL,NULL,NULL,'JohanPress5@freeweb.de',NULL,EMPTY_CLOB(),NULL,EMPTY_CLOB(),NULL,'aav4EQKbAUmgc7DQIQq7M3J5zFbgH','eKS1gqRCgpNvq',6
 282.54,'Symbouras',NULL,'Yuzhou','KU','R',NULL,'SSanders4@myspace.cn','d',EMPTY_CLOB(),'i3OtQrZ7frc',EMPTY_CLOB(),'A1EVEiFdtJlxzGp4nXPggT','wC74cXu7VvhOCTYo6gVIJtxenb',NULL,7
 5416.89,NULL,'381 Summer Drive',NULL,NULL,'qaBGyR',NULL,'MartWard3@weboffice.nl','6',EMPTY_CLOB(),'lugNFD38TcC1jQ0xiYYXwSfeMA7w',EMPTY_CLOB(),'RITHO5aFICHATbOwjD5KXtCUkc5','wWsEqtgha67Ur8MA',NULL,8
 9079.01,'Millis','9 J. Carcione Way','Abidjan','wUodsvIH6Wh3SApym1e','3Y7y2NOtYjvThoUDF5UNY','1-938-720-9219','Johan.Petrzelka@mobileme.es','8',EMPTY_CLOB(),'8iOC0z',EMPTY_CLOB(),'1mH5b','dOqahbqbrLpnxK2MII48ziV4H',NULL,9
 980088.29,'Clarke',NULL,'C □ doba',NULL,'FKVDikNXfS4DxbuxNmgVQRTI4xo','959-2667','Frank.Watson5@kpn.ca','4',EMPTY_CLOB(),'EolhS1Kyx',EMPTY_CLOB(),'Rvt7bB2M','GPR46xdLwKQMT','7u73FiaY66ALA2kfi1uKb4BprD',10

1,3
 2,4
 5,6
 8,7
 9,6
 5,8
 7,9
 1,9
 6,6
 7,7

1,11
 2,12
 3,1
 4,4
 5,6
 6,7

7,16
8,15
9,15
10,15

1,1,11
2,2,12
3,3,1
4,4,4
5,5,6
6,6,7
7,7,16
8,8,15
9,9,15
10,10,15

'Lejarette',77551.57,'2yWE','916-1488','4 W. Sunnyside Ave.','23-Mar-09','JFh3yhtW',NULL,'Outpatient',1,1,1,NULL,NULL,NULL,'Y.Slocum@mymail.net','1-102-0959','6FddicOHrl0xm1855Y2nAlj','yLwyv7A8o6PslO7Od4Kelb','Jaipur',NULL,'Deleo',315.86,'23-May-05',NULL
'Massingill',32.62,'M0iJ5hpeB0JMaVz','593-212-4508',NULL,'27-May-02',NULL,'COhQ64kDIZ6gyqnPJ7','Outpatient',3,3,3,NULL,NULL,NULL,'JHankins@freeweb.co.uk','614-911-0298','NS6d4PHQk6',NULL,'Changshu',NULL,'Paul',NULL,NULL,'03-Feb-08'
'Bitmacs',NULL,'tX4HmqqNMRvatxZYfKkfJfKAgUT','1-745-292-3871','7292 Walkers Avenue','27-Jan-01','hC',NULL,'Discharged',4,4,4,NULL,NULL,NULL,'Petra.Symbouras4@freeweb.com','1-051-286-9343',NULL,'Zsxax7IzyvazvTMNu2SyxIQC4','Chengdu','14 Reeves Street','Ijukop',92.54,'03-Apr-01','31-Dec-11'
'Korkovski',59.14,'vDd28qxwpWa6nyKNGd3fC46gxuoxnk','1-686-9951',NULL,'09-Apr-03','TbMVf7S7CJtKlvvZpFpG4lf8afZTL','2HolZadOE','Resident',5,5,5,5,5,6,'LRoger5@dolfijn.co.uk','551-1857','ZM4HV0A76D0RNHmUhCzORj','JzCGh8lrWOpQSCDrKJLcwRr5v','Delhi','2162 Shawnee Ct',NULL,7888.57,'27-Jul-09','29-Dec-03'
'Harness',4815.51,'mXvrGGLBu0jEP7A004VU6dyd6j','233-679-6764',NULL,'06-Aug-03','IB','UhYkIDCjO7ffzx','Resident',6,6,6,6,6,7,'VictorIonescu@telefonica.us','900-3961','uuyLZ','UftllyNy3TP4sC36mUEmTzfy','Yokohama','67 Elm Waterford Drive',NULL,19.71,NULL,'13-Aug-10'
NULL,NULL,'e','843-200-6778','1 East 74th Street',NULL,NULL,'t','Resident',7,8,8,7,7,16,'Bas.Crocetti@myspace.no','1-923-0138','ka7Zz','2','Havanna','948 Southwest Blvd',NULL,5.8,'14-Mar-13','07-Feb-05'
'Daley',6.46,'v2liHArINonmMPpu',NULL,'1 Glenridge Drive','11-Aug-06','xlHhbaoAmvQqFa4yqKPgjSWAkXYK','3kxQf0KVm4BYWTeKBpHmW','Resident',8,7,7,8,8,15,'Hank.Sakurai@hotmail.co.uk','367-3415',NULL,'ApJegcepRRvdHqY4','ANKARA','67 Durness Place','Wargula',5018.2,NULL,'14-Apr-12'
'Ayers',NULL,NULL,NULL,NULL,'13-Oct-06',NULL,'CuFq','Resident',9,9,9,9,9,15,NULL,'346-5200','Dbk1877s',NULL,'Macheng',NULL,NULL,8.72,NULL,'23-Mar-10'

1,'19-Jun-09',EMPTY_CLOB(),'20-Oct-02',1,1,1
3,'19-Jun-09',EMPTY_CLOB(),'01-Feb-02',3,3,3
4,'18-Feb-04',EMPTY_CLOB(),'13-Nov-01',4,4,4
5,'03-Apr-00',EMPTY_CLOB(),'01-Aug-02',5,5,5

6,'01-Feb-13',EMPTY_CLOB(),'13-Nov-01',6,6,6
7,'23-Feb-05',EMPTY_CLOB(),'23-Sep-02',7,8,8
8,'22-Jan-00',EMPTY_CLOB(),'23-Sep-12',8,7,7
9,'23-Sep-10',EMPTY_CLOB(),'23-Sep-02',9,9,9

'07-Aug-11','17-Nov-10',1,1,3,1,1
'27-Dec-11','27-Aug-07',2,3,4,3,3
'23-Aug-07','31-Oct-07',3,4,5,4,4
'06-Nov-10','06-Oct-06',4,5,6,5,5
'05-Aug-01','28-Dec-03',5,6,7,6,6
'19-Dec-03','05-Nov-10',7,7,9,8,8
'18-Nov-00','24-Sep-04',8,8,10,7,7
'25-Feb-07','17-Jul-05',9,9,2,9,9

1,'11-Mar-07',NULL,1,1,1,1,5
3,'06-May-07','29-Mar-01',3,3,3,3,7
4,'02-Jan-10','11-Jul-12',4,4,4,4,8
5,'06-May-07','24-May-05',5,5,5,5,9
6,'24-Feb-00',NULL,6,6,6,6,10
7,'07-Sep-12','03-Aug-09',7,7,8,8,1
8,'02-Jan-10','19-Feb-01',8,8,7,7,2
9,'23-Sep-04','26-May-02',9,9,9,9,3

'03-Dec-06',NULL,3.24,6,1,11,1,1,1)
'25-May-08','30-Nov-03',0583.26,8,3,1,3,3,3
NULL,'07-Sep-01',5.75,9,4,4,4,4,4
'15-Jan-12','04-Mar-03',371.18,10,5,6,5,5,5
'10-Aug-00','12-Apr-00',94.80,1,6,7,6,6,6
'18-Jan-13','18-May-03',9685.93,2,7,16,7,8,8
'20-May-09','04-Apr-07',35.59,3,8,15,8,7,7
'31-Jul-11','14-May-07',24.01,4,9,15,9,9,9

'02-Jul-11','26-Apr-01',86802.23,1,1,1,1,1
'08-Mar-10',NULL,681.42,3,3,3,3
'06-Oct-04','04-Mar-11',NULL,4,4,4,4
NULL,'10-Apr-06',9832.85,5,5,5,5
NULL,NULL,6.61,6,6,6,6
NULL,'18-Jul-05',NULL,7,7,8,8
'09-Mar-00','28-Jan-06',NULL,8,8,7,7
'08-Mar-09','19-Sep-10',490.4,9,9,9,9

1,1,1,1,1,5,1,4
3,3,3,3,3,7,3,15
4,4,4,4,4,8,4,11
5,5,5,5,5,9,5,15
6,6,6,6,6,10,6,4
7,7,7,8,8,1,7,11
8,8,8,7,7,2,8,15
9,9,9,9,9,3,9,1

'04-Sep-10','NURSE',1,11,1,11,'AnnMairy@myspace.it','011-1797',NULL,'zICk711jdyFUbFoJFdZ3u1Q','Kharkov','8 Brookfields','Troher',9913.82,'LICENSED',1,NULL
 '28-Jan-05','NURSE',2,12,2,12,'PaulinePoole@mymail.de','340-7398','bJQsUONqdN','SJhKFrCdeHPfIbdl',NULL,'9692 N. Glenoaks Blvd','Caffray',7.28,'REGISTERED',54600.51,NULL
 '29-Feb-04','NURSE',3,4,3,4,'WillPyland1@gawab.cn','1-099-8859',NULL,'yYtQKvJnXnP5wpXoI5ouD','Sendai',NULL,'Dean',43,'REGISTERED',456.94,'jQ250zyrbrOPAc'
 '20-Mar-05','NURSE',4,15,4,15,'Lynn.Phelps5@excite.org',NULL,'qzvixLYe6K3uLAiw3Ln','tcZtJzwatOqLy67E21Bx7WiHILV','Fortaleza','640 Sky Park Circle','Ahlgren',938161.25,'REGISTERED',611.13,NULL
 '29-Feb-04','NURSE',5,6,5,6,'GeoffryPengilly1@mymail.es','217-992-6075','PIRMYPHjH1R',NULL,'Linyi','1 W. Sunnyside Ave.','Poole',4,'LICENSED',69713.01,'5'
 '30-May-03','STAFF',6,7,6,7,'Nick.Poplock@mymail.ca','1-500-8885',NULL,'JUUmUloWUmbY7AjDBm5R7G','Guarulhos','8961 New London Street','Clark',3,'LICENSED',4566.94,'HMvJrILg'
 '17-Oct-05','STAFF',7,16,7,16,NULL,NULL,NULL,'LS2uD8301N',NULL,NULL,'Langham',33.16,'LICENSED',771.82,'IZm47x6mKvU65R8bGySxE0ZSIjRw'
 '09-Aug-07','STAFF',8,1,8,1,'G.Conley@telfort.us','946-1956','AdeOIkTpJN471eGrJllrrtbrn','P','Xinghua','3498 Braemar Drive','Nefos',2.18,'REGISTERED',75211.29,'UrDMmWYclmsLNub'
 '19-Sep-11','STAFF',10,1,10,1,'T.vanGoes@yahoo.ca',NULL,'B2fsv0BVqGzYtoYUS','g6NjEzjiAXeT7j03GMiJqK1Ly',NULL,'23 Chesterfield Road South','Lamere',1224.3,'REGISTERED',6700.87,'dH1NLeeAKe4YLml'

'20-Apr-11','03-Jun-12',380518,1,2,11,1,4
 '27-Apr-03','08-Mar-03',31,2,3,12,4,5
 '20-Oct-00','17-Jul-03',551,3,4,4,6,6
 '24-Sep-10','01-Apr-11',NULL,4,5,15,7,7
 '23-Oct-01','15-Dec-00',7768,5,6,6,7,8
 NULL,'18-Aug-09',7,6,7,7,11,9
 '20-Jun-03','15-Jan-07',3,7,8,16,12,10
 '02-Aug-04','13-Aug-01',NULL,8,9,1,15,1

1,NULL,EMPTY_CLOB(),5256,98,NULL,1,1,1,1,11
 3,17.03,EMPTY_CLOB(),NULL,7412,45858,3,3,3,3,4
 4,3.45,EMPTY_CLOB(),50,6,59515,4,4,4,4,15
 5,NULL,EMPTY_CLOB(),410,1780,6,5,5,5,5,6
 6,NULL,EMPTY_CLOB(),NULL,NULL,4651,6,6,6,6,7
 7,48628.28,EMPTY_CLOB(),NULL,440,NULL,7,8,8,7,16
 8,6474.58,EMPTY_CLOB(),40111,7300,NULL,8,7,7,8,1

1,11,1
 2,12,4
 3,4,11
 4,15,6
 5,6,7
 6,7,12
 7,16,15

8,1,16
10,1,7

Phase 4 Deliverables

- Web Interface
- Front End
- List of software tools
- Overall Lessons Learned

Phase IV deliverables

- 1 Develop a web-enabled interface for the Mountain View Community Hospital case study database
- 2 Write a front-end interface using one table in Oracle 10g/11i. Your front-end could be using Java or some other language. Your front-end must be username/password protected.
- 3 The following operations need to be demonstrated. Insert ,Update, Delete, Query of records from the Web-enable front-end.

HomeController.java

```
1 package com.database.finalproject;
2
3 import java.text.DateFormat;
4
5
6
7
8
9
10
11
12
13
14
15
16 /**
17  * Handles requests for the application home page.
18  */
19 @Controller
20 public class HomeController {
21
22     private static final Logger logger =
23         LoggerFactory.getLogger(HomeController.class);
24
25     /**
26      * Simply selects the home view to render by returning its name.
27      */
28     @RequestMapping(value = "/", method = RequestMethod.GET)
29     public String home(Locale locale, Model model) {
30         logger.info("Welcome home! The client locale is {}.",
31             locale);
32
33         Date date = new Date();
34         DateFormat dateFormat =
35             DateFormat.getDateInstance(DateFormat.LONG, locale);
36
37         String formattedDate = dateFormat.format(date);
38
39         model.addAttribute("serverTime", formattedDate );
40
41         return "home";
42     }
43
44     @RequestMapping(value = "/test", method = RequestMethod.GET)
45     public @ResponseBody String getPetientById(){
46
47         return "test";
48     }
49 }
```

HomeController.java

```
49 }  
50
```

DiagnosisController.java

```
1 package com.database.finalproject.controller;
2
3 import java.util.List;
20
21 @Controller
22 @RequestMapping(value = "/diagnose")
23 public class DiagnosisController {
24
25     @Autowired
26     DiagnosisDao diagnosisDao;
27
28     @Autowired
29     PatientDao patientDao;
30
31     @Autowired
32     PhysicianDXDao physicianDXDao;
33
34     @RequestMapping(value="/", method=RequestMethod.POST)
35     public String toDiagnose(@RequestParam int patientId, Model
        model){
36         model.addAttribute("patientId", patientId);
37         return "diagnose";
38     }
39
40     @RequestMapping(value="/patient/{id}", method=RequestMethod.GET)
41     public @ResponseBody String getDiagnosisByPatient(@PathVariable
        int id){
42         Gson gson = new Gson();
43         Diagnosis diagnosis = diagnosisDao.getDiagnosisByPatient(id);
44         return gson.toJson(diagnosis);
45     }
46
47     @RequestMapping(value="/diagnoseId/{id}",
        method=RequestMethod.GET)
48     public @ResponseBody String getDiagnosisById(@PathVariable int
        id){
49         Gson gson = new Gson();
50         Diagnosis diagnosis = diagnosisDao.getDiagnosisById(id);
51         return gson.toJson(diagnosis);
52     }
53
```

DiagnosisController.java

```
54     @RequestMapping(value="/diagnoseId/", method=RequestMethod.GET)
55     public @ResponseBody String getAllDiagnoses(){
56         Gson gson = new Gson();
57         List<Diagnosis> list = diagnosisDao.getDiagnoses();
58         return gson.toJson(list);
59     }
60
61
62 }
63
```

DiagnosisController.java

```
1 package com.database.finalproject.controller;
2
3 import java.util.List;
20
21 @Controller
22 @RequestMapping(value = "/diagnose")
23 public class DiagnosisController {
24
25     @Autowired
26     DiagnosisDao diagnosisDao;
27
28     @Autowired
29     PatientDao patientDao;
30
31     @Autowired
32     PhysicianDXDao physicianDXDao;
33
34     @RequestMapping(value="/", method=RequestMethod.POST)
35     public String toDiagnose(@RequestParam int patientId, Model
        model){
36         model.addAttribute("patientId", patientId);
37         return "diagnose";
38     }
39
40     @RequestMapping(value="/patient/{id}", method=RequestMethod.GET)
41     public @ResponseBody String getDiagnosisByPatient(@PathVariable
        int id){
42         Gson gson = new Gson();
43         Diagnosis diagnosis = diagnosisDao.getDiagnosisByPatient(id);
44         return gson.toJson(diagnosis);
45     }
46
47     @RequestMapping(value="/diagnoseId/{id}",
        method=RequestMethod.GET)
48     public @ResponseBody String getDiagnosisById(@PathVariable int
        id){
49         Gson gson = new Gson();
50         Diagnosis diagnosis = diagnosisDao.getDiagnosisById(id);
51         return gson.toJson(diagnosis);
52     }
53
```

DiagnosisController.java

```
54     @RequestMapping(value="/diagnoseId/", method=RequestMethod.GET)
55     public @ResponseBody String getAllDiagnoses(){
56         Gson gson = new Gson();
57         List<Diagnosis> list = diagnosisDao.getDiagnoses();
58         return gson.toJson(list);
59     }
60
61
62 }
63
```

PatientController.java

```
1 package com.database.finalproject.controller;
2
3 import java.util.Calendar;
18
19 @Controller
20 @RequestMapping(value="/patient")
21 public class PatientController {
22     private final int BEDIDFOROUT = 1;
23     private final int ROOMIDFOROUT = 1;
24     private final int FACILITYIDFOROUT = 11;
25
26     @Autowired
27     private PatientDao patientDao;
28
29     @RequestMapping(value="/", method = RequestMethod.GET)
30     public String listPatient(){
31         return "patient";
32     }
33
34
35     @RequestMapping(value= "/patientId/", method =
36     RequestMethod.GET)
37     public @ResponseBody String getAllPatients(){
38         Gson g = new Gson();
39         List<Patient> list = patientDao.getPatients();
40         return g.toJson(list);
41     }
42
43     @RequestMapping(value= "/patientId/{id}", method =
44     RequestMethod.GET)
45     public @ResponseBody String getPetientById(@PathVariable int
46     id){
47         Gson g = new Gson();
48         Patient p = patientDao.getPatient(id);
49         return g.toJson(p);
50     }
51
52     @RequestMapping(value = "/patientName/{name}", method =
53     RequestMethod.GET)
54     public @ResponseBody String getPetientByName(@PathVariable
```

PatientController.java

```
String name){
52     List<Patient> list = patientDao.getPatient(name);
53     Gson g = new Gson();
54     return g.toJson(list);
55 }
56
57 @RequestMapping(value = "/delete/{patientId}", method =
RequestMethod.GET)
58 public @ResponseBody String deletePatientById(@PathVariable int
patientId){
59     patientDao.deletePatient(patientId);
60     Gson g = new Gson();
61     List<Patient> list = patientDao.getPatients();
62     return g.toJson(list);
63 }
64
65 @RequestMapping(value = "/add", method = RequestMethod.GET)
66 public String createPatient(){
67
68     return "addPatient";
69 }
70
71 @RequestMapping(value = "/count", method = RequestMethod.GET)
72 public @ResponseBody String getPatientCount(){
73
74     return Integer.toString(patientDao.getMaxPatientId()+1);
75 }
76
77
78
79 @RequestMapping(value = "/update/{id}", method =
RequestMethod.POST)
80 public @ResponseBody void updateById(
81     @PathVariable int id,
82     @RequestParam String pType,
83     @RequestParam int pBedNo,
84     @RequestParam int pRoomNo,
85     @RequestParam int pFacilityNo)
86 {
87
88     if(pType.contentEquals("Resident")){
```


PatientController.java

```
89         patientDao.updatePatient(id, pType, pBedNo, pRoomNo,
    pFacilityNo);
90     } else {
91         patientDao.updatePatient(id, pType, BEDIDFOROUT,
    ROOMIDFOROUT, FACILITYIDFOROUT);
92     }
93 }
94
95 @RequestMapping(value = "/create", method = RequestMethod.POST)
96 public @ResponseBody void createPatient(
97     @RequestParam int patientID,
98     @RequestParam int admittingPhysician,
99     @RequestParam int bedNo,
100    @RequestParam int roomNo,
101    @RequestParam int facilityNo,
102    @RequestParam String fName,
103    @RequestParam String lName,
104    @RequestParam String type,
105    @RequestParam String city,
106    @RequestParam String address,
107    @RequestParam String state,
108    @RequestParam String phone,
109    @RequestParam String zip)
110
111
112 {
113
114
115     patientDao.addPatient(fName, lName, patientID, type,
    admittingPhysician, admittingPhysician, bedNo, roomNo, facilityNo,
    phone, city, zip, state, address);
116 }
117
118
119
120
121
122 }
```

PhysicianDXController.java

```
1 package com.database.finalproject.controller;
2
3 import java.util.Date;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 @Controller
19 @RequestMapping(value="/physicianDX")
20 public class PhysicianDXController {
21
22     @Autowired
23     PhysicianDXDao physicianDXDao;
24
25     @Autowired
26     PatientDao patientDao;
27
28     @RequestMapping(value="/add", method=RequestMethod.POST)
29     public @ResponseBody String addDiagnoseToPatient(
30         @RequestParam int patientId,
31         @RequestParam int diagnoseId,
32         @RequestParam int physicianId)
33     {
34         Patient patient = patientDao.getPatient(patientId);
35         physicianDXDao.addPhysicianDX(new Date(), new Date(),
36             physicianId, patientId, diagnoseId, patient.getPhysicianIdAdmit(),
37             patient.getReferringPhysician());
38     }
39
40
41 }
42
```

Diagnosis.java

```
1 package com.database.finalproject.model;
2
3 public class Diagnosis {
4
5     private int id;
6     private String name;
7     public int getId() {
8         return id;
9     }
10    public void setId(int id) {
11        this.id = id;
12    }
13    public String getName() {
14        return name;
15    }
16    public void setName(String name) {
17        this.name = name;
18    }
19    public Diagnosis(int id, String name) {
20        super();
21        this.id = id;
22        this.name = name;
23    }
24
25
26
27
28 }
```

DiagnosisRowMapper.java

```
1 package com.database.finalproject.model;
2
3 import java.sql.ResultSet;
4
5
6
7
8
9 @Component
10 public class DiagnosisRowMapper implements RowMapper<Diagnosis>{
11
12     @Override
13     public Diagnosis mapRow(ResultSet rs, int rowNum) throws
        SQLException {
14         // TODO Auto-generated method stub
15         return new Diagnosis(rs.getInt("DIAGNOSISID"),
            rs.getString("DIAGNOSISNAME"));
16     }
17 }
18
```

Patient.java

```
1 package com.database.finalproject.model;
2
3 public class Patient {
4
5     private int patientId;
6     private int physicianIdAdmit;
7     private int referringPhysician;
8     private int Bed;
9     private int Room;
10    private int careCenter;
11    private String fname;
12    private String lname;
13    private String address;
14    private String zip;
15    private String patientType;
16    private String dateAdmitted;
17    private String city;
18    private String state;
19    private String phone;
20    private int facilityNo;
21    public Patient(int patientId, int physicianIdAdmit, int
        referringPhysician,
22                int bed, int room, int careCenter, String fname, String
        lname,
23                String address, String zip, String patientType,
24                String dateAdmitted, String city, String state, String
        phone,
25                int facilityNo) {
26        super();
27        this.patientId = patientId;
28        this.physicianIdAdmit = physicianIdAdmit;
29        this.referringPhysician = referringPhysician;
30        Bed = bed;
31        Room = room;
32        this.careCenter = careCenter;
33        this.fname = fname;
34        this.lname = lname;
35        this.address = address;
36        this.zip = zip;
37        this.patientType = patientType;
38        this.dateAdmitted = dateAdmitted;
```

Patient.java

```
39     this.city = city;
40     this.state = state;
41     this.phone = phone;
42     this.facilityNo = facilityNo;
43 }
44 public int getPatientId() {
45     return patientId;
46 }
47 public void setPatientId(int patientId) {
48     this.patientId = patientId;
49 }
50 public int getPhysicianIdAdmit() {
51     return physicianIdAdmit;
52 }
53 public void setPhysicianIdAdmit(int physicianIdAdmit) {
54     this.physicianIdAdmit = physicianIdAdmit;
55 }
56 public int getReferringPhysician() {
57     return referringPhysician;
58 }
59 public void setReferringPhysician(int referringPhysician) {
60     this.referringPhysician = referringPhysician;
61 }
62 public int getBed() {
63     return Bed;
64 }
65 public void setBed(int bed) {
66     Bed = bed;
67 }
68 public int getRoom() {
69     return Room;
70 }
71 public void setRoom(int room) {
72     Room = room;
73 }
74 public int getCareCenter() {
75     return careCenter;
76 }
77 public void setCareCenter(int careCenter) {
78     this.careCenter = careCenter;
79 }
```

Patient.java

```
80     public String getFname() {
81         return fname;
82     }
83     public void setFname(String fname) {
84         this.fname = fname;
85     }
86     public String getLname() {
87         return lname;
88     }
89     public void setLname(String lname) {
90         this.lname = lname;
91     }
92     public String getAddress() {
93         return address;
94     }
95     public void setAddress(String address) {
96         this.address = address;
97     }
98     public String getZip() {
99         return zip;
100    }
101    public void setZip(String zip) {
102        this.zip = zip;
103    }
104    public String getPatientType() {
105        return patientType;
106    }
107    public void setPatientType(String patientType) {
108        this.patientType = patientType;
109    }
110    public String getDateAdmitted() {
111        return dateAdmitted;
112    }
113    public void setDateAdmitted(String dateAdmitted) {
114        this.dateAdmitted = dateAdmitted;
115    }
116    public String getCity() {
117        return city;
118    }
119    public void setCity(String city) {
120        this.city = city;
```

Patient.java

```
121     }
122     public String getState() {
123         return state;
124     }
125     public void setState(String state) {
126         this.state = state;
127     }
128     public String getPhone() {
129         return phone;
130     }
131     public void setPhone(String phone) {
132         this.phone = phone;
133     }
134     public int getFacilityNo() {
135         return facilityNo;
136     }
137     public void setFacilityNo(int facilityNo) {
138         this.facilityNo = facilityNo;
139     }
140
141
142
143
144
145
146 }
```


PatientRowMapper.java

```
1 package com.database.finalproject.model;
2
3 import java.sql.ResultSet;
4
5
6
7
8
9
10 @Component
11 public class PatientRowMapper implements RowMapper<Patient>{
12
13     @Override
14     public Patient mapRow(ResultSet rs, int rowNum) throws
        SQLException {
15         // TODO Auto-generated method stub
16         return new Patient(
17             rs.getInt("PATIENTID"),
18             rs.getInt("PHYSICIANIDADMIT"),
19             rs.getInt("PHYSICIANIDREFER"),
20             rs.getInt("BEDID"),
21             rs.getInt("ROOMID"),
22             rs.getInt("FACILITYID"),
23             rs.getString("PERSONFNAME"),
24             rs.getString("PERSONLNAME"),
25             rs.getString("PERSONADDRESS"),
26             rs.getString("PERSONZIP"),
27             rs.getString("PATIENTTYPE"),
28             rs.getString("DATEADMITTED"),
29             rs.getString("PERSONCITY"),
30             rs.getString("PERSONSTATE"),
31             rs.getString("PERSONPHONE"),
32             rs.getInt("FACILITYID"));
33     }
34 }
35
```

PhysicianDX.java

```
1 package com.database.finalproject.model;
2
3 import java.util.Date;
4
5 public class PhysicianDX {
6
7     private Date pDate;
8     private Date pTime;
9     private int physicianId;
10    private int patientId;
11    private int diagnosisId;
12    private int physicianIdAdmit;
13    private int physicianIdRefer;
14
15    public PhysicianDX(Date pDate, Date pTime, int physicianId, int
    patientId,
16        int diagnosisId, int physicianIdAdmit, int
    physicianIdRefer) {
17        this.pDate = pDate;
18        this.pTime = pTime;
19        this.physicianId = physicianId;
20        this.patientId = patientId;
21        this.diagnosisId = diagnosisId;
22        this.physicianIdAdmit = physicianIdAdmit;
23        this.physicianIdRefer = physicianIdRefer;
24    }
25
26    public Date getpDate() {
27        return pDate;
28    }
29
30    public void setpDate(Date pDate) {
31        this.pDate = pDate;
32    }
33
34    public Date getpTime() {
35        return pTime;
36    }
37
38    public void setpTime(Date pTime) {
39        this.pTime = pTime;
```

PhysicianDX.java

```
40     }
41
42     public int getPhysicianId() {
43         return physicianId;
44     }
45
46     public void setPhysicianId(int physicianId) {
47         this.physicianId = physicianId;
48     }
49
50     public int getPatientId() {
51         return patientId;
52     }
53
54     public void setPatientId(int patientId) {
55         this.patientId = patientId;
56     }
57
58     public int getDiagnosisId() {
59         return diagnosisId;
60     }
61
62     public void setDiagnosisId(int diagnosisId) {
63         this.diagnosisId = diagnosisId;
64     }
65
66     public int getPhysicianIdAdmit() {
67         return physicianIdAdmit;
68     }
69
70     public void setPhysicianIdAdmit(int physicianIdAdmit) {
71         this.physicianIdAdmit = physicianIdAdmit;
72     }
73
74     public int getPhysicianIdRefer() {
75         return physicianIdRefer;
76     }
77
78     public void setPhysicianIdRefer(int physicianIdRefer) {
79         this.physicianIdRefer = physicianIdRefer;
80     }
```

PhysicianDX.java

```
81  
82 }
```

database.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:p="http://www.springframework.org/schema/p"
5       xmlns:tx="http://www.springframework.org/schema/tx"
6       xmlns:c="http://www.springframework.org/schema/c"
7       xsi:schemaLocation="http://www.springframework.org/schema/
      beans http://www.springframework.org/schema/beans/spring-beans.xsd
      http://www.springframework.org/schema/tx http://
      www.springframework.org/schema/tx/spring-tx.xsd">
8
9 <!--      <bean id="dataSource"
      class="org.apache.commons.dbcp.BasicDataSource" destroy-
      method="close"> -->
10 <!--      <property name="driverClassName"
      value="oracle.jdbc.driver.OracleDriver"/> -->
11 <!--      <property name="url"
      value="jdbc:oracle:thin:@adbinstance.ckoa6pmepaak.us-
      east-1.rds.amazonaws.com:1521:adb"/> -->
12 <!--      <property name="username" value="awsuser"/> -->
13 <!--      <property name="password" value="awspassword"/> -->
14 <!--      </bean> -->
15
16      <bean id="dataSource"
      class="org.apache.commons.dbcp.BasicDataSource" destroy-
      method="close">
17      <property name="driverClassName"
      value="oracle.jdbc.driver.OracleDriver"/>
18      <property name="url"
      value="jdbc:oracle:thin:@10.88.12.4:1521/homeDB"/>
19      <property name="username" value="adb4"/>
20      <property name="password" value="adb4"/>
21  </bean>
22
23  <bean
      class="org.springframework.jdbc.core.namedparam.NamedParameterJdbcTem-
      plate"
24      c:dataSource-ref="dataSource" />
25
26  <bean class="org.springframework.jdbc.core.JdbcTemplate"
27      c:dataSource-ref="dataSource" />
```

database.xml

```
28
29     <bean id="jdbcTransactionManager"
      class="org.springframework.jdbc.datasource.DataSourceTransactionManag
      er"
30         p:dataSource-ref="dataSource"/>
31     <tx:annotation-driven transaction-
      manager="jdbcTransactionManager"/>
32
33
34 </beans>
35
```

servlet-context.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans:beans xmlns="http://www.springframework.org/schema/mvc"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:beans="http://www.springframework.org/schema/beans"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xsi:schemaLocation="http://www.springframework.org/schema/mvc
7         http://www.springframework.org/schema/mvc/spring-mvc.xsd
8         http://www.springframework.org/schema/beans http://
9         www.springframework.org/schema/beans/spring-beans.xsd
10        http://www.springframework.org/schema/context http://
11        www.springframework.org/schema/context/spring-context.xsd">
12
13     <!-- DispatcherServlet Context: defines this servlet's request-
14     processing infrastructure -->
15     <beans:import resource="database.xml"/>
16     <!-- Enables the Spring MVC @Controller programming model -->
17     <annotation-driven />
18
19     <!-- Handles HTTP GET requests for /resources/** by efficiently
20     serving up static resources in the ${webappRoot}/resources directory
21     -->
22     <resources mapping="/resources/**" location="/resources/" />
23
24     <!-- Resolves views selected for rendering by @Controllers
25     to .jsp resources in the /WEB-INF/views directory -->
26     <beans:bean
27         class="org.springframework.web.servlet.view.InternalResourceViewResol
28         ver">
29         <beans:property name="prefix" value="/WEB-INF/views/" />
30         <beans:property name="suffix" value=".jsp" />
31     </beans:bean>
32
33     <context:component-scan base-
34     package="com.database.finalproject" />
35
36 </beans:beans>
```

addPatient.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4   "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6   <!-- Requirements:
7   Query database to see if information matches a previous patient.
8   Check fname, lname, DOB.
9   List of care centers
10  List of rooms in selected care center
11  List of beds in selected room
12
13  I believe I can get this asynchronously. For the first round, I may
14  just develop using dummy data.
15  -->
16  <!-- Need to get facility, care center, and room no on the client
17  side and display accordingly.
18  Ask Zihang about creating new DAO objects for this.
19  -->
20  <head>
21    <title>
22      Add Patient
23    </title>
24    <link rel="stylesheet" type="text/css" href="../../resources/
25      css/bootstrap.css">
26    <link rel="stylesheet" type="text/css" href="../../resources/
27      css/mvch.css">
28    <script src="../../resources/js/jquery.min.js" type="text/
29      javascript"></script>
30    <script src="../../resources/js/FacilityJSON.js" type="text/
31      javascript"></script>
32    <script src="../../resources/js/AddPatient.js" type="text/
33      javascript"></script>
34  </head>
35  <body>
36
37    <script type="text/javascript">
38      var currentPhysician = '${sessionScope.physician}';
```


addPatient.jsp

```
33     </script>
34     <div id="three-column" class="container">
35         <div id="tbox1">
36             <div class="box-style-tall box-style01">
37                 <br />
38                 <div class="content">
39                     <form class="form-horizontal">
40                         <div class="control-group">
41                             <label class="control-label"
for="inputFName">First Name</label>
42                             <div class="controls">
43                                 <input type="text"
id="inputFName" placeholder="First Name">
44                             </div>
45                         </div>
46                         <div class="control-group">
47                             <label class="control-label"
for="inputLName">Last Name</label>
48                             <div class="controls">
49                                 <input type="text"
id="inputLName" placeholder="Last Name">
50                             </div>
51                         </div>
52                         <div class="control-group">
53                             <label class="control-label"
for="inputPatientID">PatientID</label>
54                             <div class="controls">
55                                 <input type="text"
id="inputPatientID" placeholder="PatientID" readonly>
56                             </div>
57                         </div>
58                         <div class="control-group">
59                             <label class="control-label"
for="inputAdmittingPhysician">Admitting Physician</label>
60                             <div class="controls">
61                                 <input type="text"
id="inputAdmittingPhysician" placeholder="Admitting Physician"
readonly>
62                             </div>
63                         </div>
64                     </form>
```

addPatient.jsp

```

65         </div>
66     </div>
67 </div>
68 <div id="tbody2">
69     <div class="box-style-tall box-style02"><br />
70         <div class="content">
71             <form class="form-horizontal">
72                 <div class="control-group">
73                     <label class="control-label"
for="Address">Address</label>
74                         <div class="controls">
75                             <input type="text"
id="inputAddress" placeholder="Street Address">
76                             </div>
77                         </div>
78                     <div class="control-group">
79                         <label class="control-label"
for="inputCity">City</label>
80                             <div class="controls">
81                                 <input type="text"
id="inputCity" placeholder="City">
82                                 </div>
83                             </div>
84                     <div class="control-group">
85                         <label class="control-label"
for="inputState">State</label>
86                             <div class="controls">
87                                 <!-- <input type="text"
id="inputState" placeholder="State"> -->
88                                 <select id="inputState">
89                                     <option value="---">
90                                         ---
91                                     </option>
92                                     <option value="AL">
93                                         Alabama
94                                     </option>
95                                     <option value="AK">
96                                         Alaska
97                                     </option>
98                                     <option value="AZ">
99                                         Arizona

```

addPatient.jsp

```
100      </option>
101      <option value="AR">
102          Arkansas
103      </option>
104      <option value="CA">
105          California
106      </option>
107      <option value="CO">
108          Colorado
109      </option>
110      <option value="CT">
111          Connecticut
112      </option>
113      <option value="DE">
114          Delaware
115      </option>
116      <option value="DC">
117          District Of Columbia
118      </option>
119      <option value="FL">
120          Florida
121      </option>
122      <option value="GA">
123          Georgia
124      </option>
125      <option value="HI">
126          Hawaii
127      </option>
128      <option value="ID">
129          Idaho
130      </option>
131      <option value="IL">
132          Illinois
133      </option>
134      <option value="IN">
135          Indiana
136      </option>
137      <option value="IA">
138          Iowa
139      </option>
140      <option value="KS">
```

addPatient.jsp

```
141         Kansas
142     </option>
143     <option value="KY">
144         Kentucky
145     </option>
146     <option value="LA">
147         Louisiana
148     </option>
149     <option value="ME">
150         Maine
151     </option>
152     <option value="MD">
153         Maryland
154     </option>
155     <option value="MA">
156         Massachusetts
157     </option>
158     <option value="MI">
159         Michigan
160     </option>
161     <option value="MN">
162         Minnesota
163     </option>
164     <option value="MS">
165         Mississippi
166     </option>
167     <option value="MO">
168         Missouri
169     </option>
170     <option value="MT">
171         Montana
172     </option>
173     <option value="NE">
174         Nebraska
175     </option>
176     <option value="NV">
177         Nevada
178     </option>
179     <option value="NH">
180         New Hampshire
181     </option>
```

addPatient.jsp

```
182      <option value="NJ">
183          New Jersey
184      </option>
185      <option value="NM">
186          New Mexico
187      </option>
188      <option value="NY">
189          New York
190      </option>
191      <option value="NC">
192          North Carolina
193      </option>
194      <option value="ND">
195          North Dakota
196      </option>
197      <option value="OH">
198          Ohio
199      </option>
200      <option value="OK">
201          Oklahoma
202      </option>
203      <option value="OR">
204          Oregon
205      </option>
206      <option value="PA">
207          Pennsylvania
208      </option>
209      <option value="RI">
210          Rhode Island
211      </option>
212      <option value="SC">
213          South Carolina
214      </option>
215      <option value="SD">
216          South Dakota
217      </option>
218      <option value="TN">
219          Tennessee
220      </option>
221      <option value="TX">
222          Texas
```

addPatient.jsp

```
223         </option>
224         <option value="UT">
225             Utah
226         </option>
227         <option value="VT">
228             Vermont
229         </option>
230         <option value="VA">
231             Virginia
232         </option>
233         <option value="WA">
234             Washington
235         </option>
236         <option value="WV">
237             West Virginia
238         </option>
239         <option value="WI">
240             Wisconsin
241         </option>
242         <option value="WY">
243             Wyoming
244         </option>
245     </select>
246
247
248
249         </div>
250     </div>
251     <div class="control-group">
252         <label class="control-label"
for="inputZipcode">Zipcode</label>
253         <div class="controls">
254             <input type="text"
id="inputZipcode" placeholder="Zipcode">
255         </div>
256     </div>
257     <div class="control-group">
258         <label class="control-label"
for="inputPhone">Phone</label>
259         <div class="controls">
260             <input type="text"
```

addPatient.jsp

```

261         id="inputPhone" placeholder="Phone">
262             </div>
263         </div>
264     </form>
265 </div>
266 </div>
267 <div id="tbox3">
268     <div class="box-style-tall box-style03"><br />
269     <div class="content">
270         <form class="form-horizontal">
271             <div class="control-group">
272                 <label class="control-label"
for="inputPatientType">Patient Type</label>
273                 <div class="controls">
274                     <label class="radio"><label
class="radio"><input type="radio" name="optionsRadios"
id="residentPatient" value="resident" checked> Resident Patient</
label> <label class="radio"><input type="radio" name="optionsRadios"
id="outpatient" value="outpatient" checked> Outpatient</label>
<label class="radio"><input type="radio" name="optionsRadios"
id="discharged" value="discharged"> Discharged</label></label>
275                     </div>
276                 </div>
277                 <span id="residentControls" >
278                 <div class="control-group">
279                     <label class="control-label"
for="inputCareCenter">Care Center</label>
280                     <div class="controls">
281                         <select id="facility">
282                             </select>
283                     </div>
284                 </div>
285                 <div class="control-group">
286                     <label class="control-label"
for="inputRoomNo">Room No.</label>
287                     <div class="controls">
288                         <select id="room">
289                             </select> <!-- <input
290 type="text" id="inputRoomNo" placeholder="Room No."> -->

```

addPatient.jsp

```
291         </div>
292     </div>
293     <div class="control-group">
294         <label class="control-label"
for="inputBedNo">Bed No.</label>
295         <div class="controls">
296             <select id="bed">
297                 </select>
298             </div>
299         </div>
300     </span>
301 </form>
302 <div class="button-box" >
303     <form>
304         <button type="button" class="btn
btn-block btn-danger" onclick="postNewPatient()">Add New Patient</
button>
305     </form>
306     <form action="../patient/">
307         <button class="btn btn-block">View
All Patients</button>
308     </form>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </body>
315 </html>
```


diagnose.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4   "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6 <head>
7   <meta http-equiv="Content-Type" content="text/html;
8     charset=ISO-8859-1">
9   <title>
10     Diagnose Patient
11   </title>
12   <script src="../../resources/js/jquery.min.js" type="text/
13     javascript">
14   </script>
15   <link rel="stylesheet" type="text/css" href="../../resources/
16     css/bootstrap.css">
17   <link rel="stylesheet" type="text/css" href="../../resources/
18     css/mvch.css">
19   <script src="../../resources/js/DiagnoseParser.js" type="text/
20     javascript"></script>
21
22 </head>
23 <body>
24   <script type="text/javascript">
25     var currentPatientId = '${patientId}';
26     var currentPhysician = '${sessionScope.physician}';
27   </script>
28
29   <div id="wrapper">
30 <!-- start top -->
31     <div id="upper-container" class="container">
32       <input id="searchBox" class="rounded" size="400"
33         placeholder=" search treatment name or number">
34     </div><br />
35 <!-- start middle -->
36     <div id="middle-container" class="container">
37       <table class="fixed">
38         <tr>
39           <td>
40             <table class="fixed">
```

diagnose.jsp

```

35         <tr>
36             <th class="title">
37                 Diagnosis
38             </th>
39             <th class="title">
40                 Diagnosis Number
41             </th>
42         </tr>
43     </table>
44 </td>
45 </tr>
46 <tr>
47     <td>
48         <div style="width:1200px; height:300px;
overflow:auto;">
49             <table class="table-hover table-
condensed table-bordered fixed">
50                 <tbody id="dataTableBody">
51                     <!-- orders will be inserted
into rows here -->
52                 </tbody>
53             </table>
54         </div>
55     </td>
56 </tr>
57 </table>
58 </div>
59 <!-- start bottom -->
60     <div id="three-column" class="container">
61         <div id="tbox1">
62             <div class="box-style box-style01"
display="height:250">
63                 <div class="content">
64                     <h2 class="title">
65                         Patient Information:
66                     </h2>
67                     <ul id="personal">
68                         <li id="name">patient name
69                     </li>
70                         <li id="phone">patient phone
71                     </li>

```

diagnose.jsp

```

72         <li id="address">patient address
73     </li>
74     <li id="cityStateZip">patient zip
75 </li>
76 </ul>
77 <ul id="patient">
78     <li id="patientId">patientID
79 </li>
80     <li
81 id="referringPhysician">referring physician
82 </li>
83     <li id="admissionType">admission
84 type
85 </li>
86 </ul>
87 </div>
88 </div>
89 <div id="tbox2">
90     <div class="box-style box-style02"
91 style="display:none" >
92         <div class="content">
93             <br />
94             <br />
95             <br />
96             <br />
97             <br />
98             <br /><br />
99         </div>
100     </div>
101     <div id="tbox3">
102         <div class="box-style box-style03"
103 display="height:250">
104             <div class="content">
105                 <h3 class="title">
106                     Action:
107                 </h3>
108                 <br /><br />
109                 <div class="button-box">
110                     <button id="diagnoseButton">

```

diagnose.jsp

```
109     class="btn btn-danger btn-block" onclick="diagnosePatient()"
110     disabled>Diagnose Patient</button>
111
112         <form action="../patient/">
113             <button id="nav-patients"
114             class="btn btn-block">Back to All Patients</button>
115         </form>
116         <br /><br />
117     </div>
118 </div>
119 </div><!-- end bottom -->
120 </body>
121 </html>
```

home.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4   "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6   <!-- requirements:
7     1) query physician table. Return true or false to tell whether logged
8       in or not.
9     If true, return physician physician information.
10   -->
11   <head>
12     <title>
13       Physician Login: MVCH
14     </title>
15     <link rel="stylesheet" type="text/css" href="../resources/css/
16       bootstrap.css">
17     <link rel="stylesheet" type="text/css" href="../resources/css/
18       mvch.css">
19     <script src="../resources/js/jquery.min.js" type="text/
20       javascript">
21   </script>
22   </head>
23   <body>
24 <section class="login-container">
25   <div class="login">
26
27     <h1>Login to Web App</h1>
28     <error>${message}</error>
29     <form method="post" action="patient/">
30       <p><input type="text" name="physicianId" value=""
31         placeholder="physician id"></p>
32       <p><input type="password" name="password" value=""
33         placeholder="password"></p>
34       <p class="submit"><input type="submit" name="commit"
35         value="Login"></p>
36     </form>
37   </div>
38 </section>
39 </body>
```

home.jsp

34 </html>

patient.jsp

```
1 <!-- Requirements:
2
3 Update Patient and/or Person Table. (will do patient information
   first.)
4
5 Lots of information on this page...
6 q: json for of all patient fname, lname, and patientID
7 q: specific information from from the patient, person, and physician
   tables
8
9 PERSON_t: (associated with patient)
10     Person_First_Name
11     Person_Last_Name
12     Person_Address
13     Person_City
14     Person_Phone
15     Person_State
16     Person_Zip
17     Person_BirthDate
18
19 Patient_t:
20     Patient_ID
21     Contact_Date
22     Contact_Time
23     Person_ID
24     Physician_ID
25     Referral_ID
26
27 Physician_t (associated with physician_id)
28     Physician_ID
29
30 Person_T (associated with physician)
31     Person_First_Name
32     Person_Last_Name
33
34 -->
35 <html>
36     <head>
37         <title>
38             Search Patients
39         </title>
```

patient.jsp

```
40     <link rel="stylesheet" type="text/css" href="../../resources/
css/bootstrap.css">
41     <link rel="stylesheet" type="text/css" href="../../resources/
css/mvch.css">
42     <script src="../../resources/js/jquery.min.js" type="text/
javascript">
43 </script>
44 <!--     <script src="../../resources/js/PatientID.js" type="text/
javascript"> -->
45 <!-- </script> -->
46     <script src="../../resources/js/FacilityJSON.js" type="text/
javascript"></script>
47     <script src="../../resources/js/PatientParser.js" type="text/
javascript">
48 </script>
49 </head>
50 <body>
51     <div id="wrapper">
52         <!-- start top -->
53         <div id="upper-container" class="container">
54             <input id="searchBox" class="rounded" size="400"
placeholder=" search patient ID, first or last name">
55         </div><!-- start middle -->
56         <div id="middle-container" class="container">
57             <table class="fixed">
58                 <tr>
59                     <td>
60                         <table class="fixed">
61                             <tr>
62                                 <th class="title">
63                                     First
64                                 </th>
65                                 <th class="title">
66                                     Last
67                                 </th>
68                                 <th class="title">
69                                     Date Admitted
70                                 </th>
71                                 <th class="title">
72                                     PatientID
73                                 </th>
```


patient.jsp

```

74         </tr>
75     </table>
76 </td>
77 </tr>
78 <tr>
79     <td>
80         <div style="width:1200px; height:200px;
overflow:auto;">
81             <table class="table-hover table-
condensed table-bordered fixed">
82                 <tbody id="dataTableBody">
83                     <!-- patients will be
inserted into rows here -->
84                 </tbody>
85             </table>
86         </div>
87     </td>
88 </tr>
89 </table>
90 </div><!-- start bottom -->
91 <div id="three-column" class="container">
92     <div id="tbox1">
93         <div class="box-style-medium box-style01">
94             <div class="content">
95                 <br>
96                 <form class="form-horizontal">
97                     <div class="control-group">
98                         <label class="control-label"
for="name">Name</label>
99                             <div class="controls">
100                                 <input type="text" id="name"
placeholder="Name" readonly>
101                             </div>
102                         </div>
103                     <div class="control-group">
104                         <label class="control-label"
for="Phone">Phone</label>
105                             <div class="controls">
106                                 <input type="text"
id="phone" placeholder="Phone">
107                             </div>

```

patient.jsp

```
108         </div>
109         <div class="control-group">
110             <label class="control-label"
111                 for="Address">Address</label>
112             <div class="controls">
113                 <input type="text"
114                     id="address" placeholder="Address">
115             </div>
116         </div>
117         <div class="control-group">
118             <label class="control-label"
119                 for="city">city</label>
120             <div class="controls">
121                 <input type="text" id="city"
122                     placeholder="City" readonly>
123             </div>
124         </div>
125         <div class="control-group">
126             <label class="control-label"
127                 for="State">State</label>
128             <div class="controls">
129                 <select id="state">
130                     <option value="---">
131                         ---
132                     </option>
133                     <option value="AL">
134                         Alabama
135                     </option>
136                     <option value="AK">
137                         Alaska
138                     </option>
139                     <option value="AZ">
140                         Arizona
141                     </option>
142                     <option value="AR">
143                         Arkansas
144                     </option>
145                     <option value="CA">
146                         California
147                     </option>
148                     <option value="CO">
```

patient.jsp

```
144         Colorado
145     </option>
146     <option value="CT">
147         Connecticut
148     </option>
149     <option value="DE">
150         Delaware
151     </option>
152     <option value="DC">
153         District Of Columbia
154     </option>
155     <option value="FL">
156         Florida
157     </option>
158     <option value="GA">
159         Georgia
160     </option>
161     <option value="HI">
162         Hawaii
163     </option>
164     <option value="ID">
165         Idaho
166     </option>
167     <option value="IL">
168         Illinois
169     </option>
170     <option value="IN">
171         Indiana
172     </option>
173     <option value="IA">
174         Iowa
175     </option>
176     <option value="KS">
177         Kansas
178     </option>
179     <option value="KY">
180         Kentucky
181     </option>
182     <option value="LA">
183         Louisiana
184     </option>
```

patient.jsp

```
185      <option value="ME">
186          Maine
187      </option>
188      <option value="MD">
189          Maryland
190      </option>
191      <option value="MA">
192          Massachusetts
193      </option>
194      <option value="MI">
195          Michigan
196      </option>
197      <option value="MN">
198          Minnesota
199      </option>
200      <option value="MS">
201          Mississippi
202      </option>
203      <option value="MO">
204          Missouri
205      </option>
206      <option value="MT">
207          Montana
208      </option>
209      <option value="NE">
210          Nebraska
211      </option>
212      <option value="NV">
213          Nevada
214      </option>
215      <option value="NH">
216          New Hampshire
217      </option>
218      <option value="NJ">
219          New Jersey
220      </option>
221      <option value="NM">
222          New Mexico
223      </option>
224      <option value="NY">
225          New York
```

patient.jsp

```
226      </option>
227      <option value="NC">
228          North Carolina
229      </option>
230      <option value="ND">
231          North Dakota
232      </option>
233      <option value="OH">
234          Ohio
235      </option>
236      <option value="OK">
237          Oklahoma
238      </option>
239      <option value="OR">
240          Oregon
241      </option>
242      <option value="PA">
243          Pennsylvania
244      </option>
245      <option value="RI">
246          Rhode Island
247      </option>
248      <option value="SC">
249          South Carolina
250      </option>
251      <option value="SD">
252          South Dakota
253      </option>
254      <option value="TN">
255          Tennessee
256      </option>
257      <option value="TX">
258          Texas
259      </option>
260      <option value="UT">
261          Utah
262      </option>
263      <option value="VT">
264          Vermont
265      </option>
266      <option value="VA">
```

patient.jsp

```

267             Virginia
268         </option>
269         <option value="WA">
270             Washington
271         </option>
272         <option value="WV">
273             West Virginia
274         </option>
275         <option value="WI">
276             Wisconsin
277         </option>
278         <option value="WY">
279             Wyoming
280         </option>
281     </select>
282 </div>
283 </div>
284 <div class="control-group">
285     <label class="control-label"
for="zip">zip</label>
286     <div class="controls">
287         <input type="text" id="zip"
placeholder="Zipcode">
288     </div>
289 </div>
290 </form>
291 </div>
292 </div>
293 </div>
294 <div id="tbox2">
295     <div class="box-style-medium box-style02">
296         <div class="content">
297             <br>
298             <form class="form-horizontal">
299                 <div class="control-group">
300                     <label class="control-label"
for="patientID">Patient ID</label>
301                     <div class="controls">
302                         <input type="text"
id="patientID" placeholder="patientID" readonly>
303                     </div>

```

patient.jsp

```

304         </div>
305         <div class="control-group">
306             <label class="control-label"
for="referringPhysician">Referring Physician</label>
307             <div class="controls">
308                 <input type="text"
id="referringPhysician" placeholder="Referring Physician">
309             </div>
310         </div>
311         <div class="control-group">
312             <label class="control-label"
for="inputPatientType">Patient Type</label>
313             <div class="controls">
314                 <label class="radio"><input
type="radio" name="optionsRadios" id="residentPatient"
value="resident"> Resident Patient</label>
315                 <label class="radio"><input
type="radio" name="optionsRadios" id="outpatient"
value="outpatient"> Outpatient</label> <label class="radio"><input
type="radio" name="optionsRadios" id="discharged" value="discharged"
checked> Discharged</label>
316             </div>
317         </div>
318         <span id="residentControls" >
319             <div class="control-group">
320                 <label class="control-label"
for="inputFacility">Facility</label>
321                 <div class="controls">
322                     <select id="facility">
323                         <option value="0">
324                             ---
325                         </option><!-- care
centers go here -->
326                     </select>
327                 </div>
328             </div>
329             <div class="control-group">
330                 <label class="control-label"
for="inputRoomNo">Room No.</label>
331                 <div class="controls">
332                     <select id="room">

```

patient.jsp

```

333                                     <option value="0">
334                                     ---
335                                     </option><!-- rooms
    go here -->
336                                     </select>
337                                     </div>
338                                     </div>
339                                     <div class="control-group">
340                                     <label class="control-label"
    for="inputBedNo">Bed No.</label>
341                                     <div class="controls">
342                                     <select id="bed">
343                                     <option value="0">
344                                     ---
345                                     </option><!-- beds
    go here -->
346                                     </select>
347                                     </div>
348                                     </div>
349                                     </span>
350                                     </form>
351                                     </div>
352                                     </div>
353                                     </div>
354                                     <div id="tbox3">
355                                     <div class="box-style-medium box-style03">
356                                     <div class="content">
357                                     <br>
358                                     <h3>
359                                     Available Actions:
360                                     </h3><br>
361                                     <br>
362                                     <div class="button-box">
363                                     <button class="btn btn-danger btn-
    block" onclick="postUpdate()">Update Patient</button>
364                                     <form action="./add">
365                                     <button class="btn btn-
    block">Add New Patient</button>
366                                     </form>
367                                     <form action="../diagnose/"
    id="diagnose" method="POST">

```


patient.jsp

```
368             <button class="btn btn-  
    block">Diagnose Patient</button>  
369             </form>  
370             <button id="deleteButton"  
    class="btn btn-block" onClick="deletePatient()" disabled>  
371                 Delete Patient</button>  
372             <form action="..">  
373                 <button class="btn btn-  
    block">Logout</button>  
374             </form>  
375         </div>  
376     </div>  
377 </div>  
378 </div>  
379 </div>  
380 </div><!-- end bottom -->  
381 </body>  
382 </html>
```

AccountDao.java

```
1 package com.database.finalproject.dao;
2
3 import java.sql.Connection;
13
14
15 @Repository("accountDao")
16 public class AccountDao {
17
18     @Inject
19     private DataSource dataSource;
20
21     public int verify(String username, String password) throws
        SQLException{
22         Connection conn = null;
23
24         conn = dataSource.getConnection();
25         String sql = "SELECT PHYSICIANID FROM ACCOUNT WHERE
        PHYSICIANID='"+username+"' AND PASSWORD = '"+ password+"'";
26         PreparedStatement preStatement = conn.prepareStatement(sql);
27         ResultSet result = preStatement.executeQuery();
28         if(result.next())
29             return result.getInt("PHYSICIANID");
30         else
31             return -1;
32     }
33 }
34
35 }
36
```

DiagnosisDao.java

```
1 package com.database.finalproject.dao;
2
3 import java.util.HashMap;
16
17 @Repository("diagnosisDao")
18 public class DiagnosisDao {
19
20     @Inject private NamedParameterJdbcOperations jdbcTemplate;
21     @Inject private DiagnosisRowMapper diagnosisRowMapper;
22
23     public List<Diagnosis> getDiagnoses(){
24         String sql = "SELECT * FROM Diagnosis";
25         return jdbcTemplate.query(sql, new HashMap<String, Object>(),
diagnosisRowMapper);
26     }
27
28     public Diagnosis getDiagnosisById(int diagnosisId){
29         String sql = "SELECT * FROM Diagnosis WHERE DiagnosisId
= :diagnosisId";
30         SqlParameterSource params = new
MapSqlParameterSource().addValue("diagnosisId", diagnosisId);
31         List<Diagnosis> list = jdbcTemplate.query(sql, params,
diagnosisRowMapper);
32         return list.isEmpty()? null : list.get(0);
33     }
34
35     public Diagnosis getDiagnosisByPatient(int patientId){
36         String sql = "SELECT DIAGNOSISID, DIAGNOSISNAME FROM
PHYSICIANDX,DIAGNOSIS WHERE PHYSICIANDX.DIAGNOSISID =
DIAGNOSIS.DIAGNOSISID AND PHYSICIANDX.patientId = :patientId";
37         SqlParameterSource params = new
MapSqlParameterSource().addValue("patientId", patientId);
38         List<Diagnosis> list = jdbcTemplate.query(sql, params,
diagnosisRowMapper);
39         return list.isEmpty()? null : list.get(0);
40
41     }
42 }
43
```

PatientDao.java

```
1 package com.database.finalproject.dao;
2
3 import java.sql.SQLException;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 @Repository("patientDao")
26 public class PatientDao {
27
28     private final Calendar calendar = Calendar.getInstance();
29
30     @Inject private NamedParameterJdbcOperations jdbcTemplate;
31     @Inject private PatientRowMapper patientRowMapper;
32     @Inject private JdbcTemplate _jdbJdbcTemplate;
33
34     public List<Patient> getPatients() {
35         String sql = "SELECT PERSONPHONE, PATIENTID,
36             PHYSICIANIDADMIT, PHYSICIANIDREFER, BEDID, ROOMID, FACILITYID,"+
37             "PERSONFNAME, PERSONLNAME, PERSONADDRESS, PERSONZIP,
38             PATIENTTYPE, DATEADMITTED, PERSONCITY, PERSONSTATE, FACILITYID FROM
39             PATIENT ";
40         return jdbcTemplate.query(sql, new HashMap<String,
41             Object>(), patientRowMapper);
42     }
43
44     public int getMaxPatientId(){
45         return _jdbJdbcTemplate.queryForInt("SELECT MAX(PatientId)
46             from PATIENT");
47     }
48
49     public Patient getPatient(int patientId) {
50         String sql = "SELECT PERSONPHONE, PATIENTID,
51             PHYSICIANIDADMIT, PHYSICIANIDREFER, BEDID, ROOMID, FACILITYID,"+
52             "PERSONFNAME, PERSONLNAME, PERSONADDRESS, PERSONZIP,
53             PATIENTTYPE, DATEADMITTED, PERSONCITY, PERSONSTATE, FACILITYID FROM
54             PATIENT WHERE PATIENTID = :patientId";
55         SqlParameterSource params = new
56             MapSqlParameterSource().addValue("patientId", patientId);
57     }
58 }
```

PatientDao.java

```
49         List<Patient> patients = jdbcTemplate.query(sql, params,
patientRowMapper);
50         return patients.isEmpty()? null : patients.get(0);
51     }
52 }
53
54     public List<Patient> getPatient(String name) {
55         String sql = "SELECT PATIENTID, PHYSICIANIDADMIT,
PHYSICIANIDREFER, BEDID, ROOMID, FACILITYID, PatientType " +
56             "FROM PATIENT WHERE PERSONNAME = :name";
57         SqlParameterSource params = new
MapSqlParameterSource().addValue("name", name);
58         List<Patient> patients = jdbcTemplate.query(sql, params,
patientRowMapper);
59         return patients;
60     }
61
62     public void updatePatient(int patientId, String type) {
63         String sql = "update Patient set PatientType = :type where
PatientId = :patientId";
64         SqlParameterSource params = new MapSqlParameterSource()
65             .addValue("type", type)
66             .addValue("patientId", patientId);
67         jdbcTemplate.update(sql, params);
68     }
69
70     public void updatePatient(int patientId, String type, int bedId,
int roomId, int facilityId) {
71         String sql = "update Patient set PatientType = :type, BedId
= :bedId, RoomId = :roomId, FacilityId = :facilityId where PatientId
= :patientId";
72         SqlParameterSource params = new MapSqlParameterSource()
73             .addValue("bedId", bedId)
74             .addValue("roomId", roomId)
75             .addValue("facilityId", facilityId)
76             .addValue("type", type)
77             .addValue("patientId", patientId);
78         jdbcTemplate.update(sql, params);
79     }
80
81     public void deletePatient(int patientId){
```

PatientDao.java

```
82     _jdbcJdbcTemplate.update("DELETE FROM PHYSICIANDX WHERE
PATIENTID = "+patientId);
83     _jdbcJdbcTemplate.update("DELETE FROM PATIENT WHERE PATIENTID
= "+patientId);
84     }
85
86     public void addPatient(String patientLName, String patientFName,
int patientId, String patientType, int physicianIdAdmit, int
physicianIdRefer,
87         int bedId, int roomId, int facilityId, String
personPhone, String personCity, String personZip, String
personState, String personAddress) {
88 //         String sql = "insert into PATIENT " +
89 //             "(PATIENTID, PHYSICIANIDADMIT, PHYSICIANIDREFER,
BEDID, ROOMID, FACILITYID, PATIENTTYPE) " +
90 //             "values
( :patientId, :physicianIdAdmit,:physicianIdRefer, :bedId, :roomId,
:facilityId, :status)";
91 //         String sql = "insert into Patient" +
92 //             "(PATIENTID, PATIENTTYPE, physicianidadmit,
physicianIdRefer, BedId, RoomId, FacilityId, PersonPhone,
PersonCity, PersonZip, PersonState, PersonAddress, DateAdmitted)" +
93 //             "values
(:patientId, :patientType, :physicianIdAdmit, :physicianIdRefer, :be
dId, :roomId, :facilityId, :personPhone, :personCity, :personZip, :p
ersonState, :personAddress, to_date(':dateAdmitted', 'mm/dd/
yyyy'))";
94 //
95         calendar.setTime(new Date());
96
97         String dateAdmitted =(calendar.get(Calendar.MONTH) + 1) +
98             "/" + calendar.get(Calendar.DAY_OF_MONTH) + "/" +
calendar.get(Calendar.YEAR);
99
100        String timeAdmitted = calendar.get(Calendar.HOUR_OF_DAY) +
":" + calendar.get(Calendar.MINUTE) + ":" +
101            calendar.get(Calendar.SECOND);
102
103        String date = dateAdmitted + ":" + timeAdmitted;
104
105        String sql = "insert into Patient"+
```

PatientDao.java

```
106         "(PERSONFNAME, PERSONLNAME, PATIENTID,
PATIENTTYPE, PHYSICIANIDADMIT,PHYSICIANIDREFER, BEDID, ROOMID,
FACILITYID, PERSONPHONE, PERSONCITY, PERSONZIP, PERSONSTATE,
PERSONADDRESS, DATEADMITTED)"+
107         "values
(:patientFName, :patientLName, :patientId, :patientType, :physicianI
dAdmit, :physicianIdRefer, :bedId, :roomId, :facilityId, :personPhon
e, :personCity, :personZip, :personState, :personAddress,
to_date(:dateAdmitted, 'mm/dd/yyyy:HH24:MI:SS'))";
108
109         SqlParameterSource params = new MapSqlParameterSource()
110         .addValue("patientId", patientId)
111         .addValue("patientFName", patientFName)
112         .addValue("patientLName", patientLName)
113         .addValue("patientType", patientType)
114         .addValue("physicianIdAdmit", physicianIdAdmit)
115         .addValue("physicianIdRefer", physicianIdRefer)
116         .addValue("bedId", bedId)
117         .addValue("roomId", roomId)
118         .addValue("facilityId", facilityId)
119         .addValue("personPhone", personPhone)
120         .addValue("personCity", personCity)
121         .addValue("personZip", personZip)
122         .addValue("personState", personState)
123         .addValue("personAddress", personAddress)
124         .addValue("dateAdmitted", date);
125         jdbcTemplate.update(sql, params);
126
127     }
128
129
130 }
131
```

PhysicianDXDao.java

```
1 package com.database.finalproject.dao;
2
3 import java.util.Calendar;
12
13 @Repository("physicianDXDao")
14 public class PhysicianDXDao {
15
16     @Inject private NamedParameterJdbcOperations jdbcTemplate;
17     private static final Calendar calendar = Calendar.getInstance();
18
19     public void addPhysicianDX(Date pDate, Date pTime, int
        physicianId, int patientId,
20         int diagnosisId, int physicianIdAdmit, int
        physicianIdRefer) {
21
22         calendar.setTime(pDate);
23         String pDateString =(calendar.get(Calendar.MONTH) + 1) +
24             "/" + calendar.get(Calendar.DAY_OF_MONTH) + "/" +
        calendar.get(Calendar.YEAR);
25
26
27
28         String sql = "INSERT INTO PhysicianDX values (" +
29             "to_date(:dateString, 'mm/dd/yyyy'), " +
30             "to_date(:dateString, 'mm/dd/yyyy')" +
31             ", :physicianId, :patientId, :diagnosisId, :physician
        IdAdmit, :physicianIdRefer)";
32
33         SqlParameterSource params = new MapSqlParameterSource()
34             .addValue("dateString", pDateString)
35             .addValue("timeString", pDateString)
36             .addValue("physicianId", physicianId)
37             .addValue("patientId", patientId)
38             .addValue("diagnosisId", diagnosisId)
39             .addValue("physicianIdAdmit", physicianIdAdmit)
40             .addValue("physicianIdRefer", physicianIdRefer);
41
42         jdbcTemplate.update(sql, params);
43
44     }
45
```


PhysicianDXDao.java

```
46 }  
47
```

```
1 facilities = {}
2
3 $(document).ready ->
4   do initialize
5
6
7 initialize = ->
8   $('#residentControls').hide()
9   $('#discharged').click(hideResidentControls)
10  $('#outpatient').click(hideResidentControls)
11  $('#resident').click(showResidentControls)
12
13  $('#facility').change ->
14    do findRooms
15
16  $('#room').change ->
17    do findBeds
18
19  do findFacilities
20
21
22 hideResidentControls = ->
23   $('#residentControls').hide()
24
25   $('#room').html('');
26   $('#room').append('<option value="0">----</option>')
27
28   $('#bed').html('');
29   $('#bed').append('<option value="0">----</option>')
30
31 showResidentControls = ->
32   $('#residentControls').show()
33   $('#facility').val(0) #default value is zero
34
35 findFacilities = ->
36   for key, value of residency
37     facilities[value.facilityNo] = value.facilityName
38
39   for key, value of facilities
40     $('#facility').append('<option value="'+key+'">'+value+'</option>')
41
42   do findRooms
43
44 findRooms = ->
45   $('#room').html('');
46   tempRooms = {}
47
48   selectedOption = parseInt($('#facility option:selected').attr('value'))
49   for key, value of residency
50     if value.facilityNo is selectedOption
51       tempRooms[value.roomNo]=value.roomNo
52       #hacky. using key-value so that value will overwrite duplicates automatically
53
54   for key, value of tempRooms
55     $('#room').append('<option value="'+key+'">'+value+'</option>')
56
57   do findBeds
58
59
60 findBeds = ->
61   $('#bed').html('');
62   tempBeds = {}
63
```

```
64     selectedOption = parseInt($('#room option:selected').attr('value'))
65
66     for key, value of residency
67       if value.roomNo is selectedOption
68         tempBeds[value.bedNo]=value.bedNo
69         #hacky. using key-value so that value will overwrite duplicates automatically
70
71     for key, value of tempBeds
72       $('#bed').append('<option value="'+key+'">'+value+'</option>')
73
74
75
76
77 postNewPatient = ->
78   newPatient = {}
79   fName = $('#inputFName').val()
80   lName = $('#inputLName').val()
81   patientID = '54902' #need to autogenrate a patient ID here that doesn't conflict. Shou
82   admittingPhysician = '439023' #should be passed from login page
83
84   address = $('#inputAddress').val()
85   city = $('#inputCity').val()
86   state = $('#inputState').val()
87   zip = $('#inputZipcode').val()
88   phone = $('#inputPhone').val()
89
90   type = null
91   facilityNo = 11
92   roomNo = 1
93   bedNo = 1
94
95
96   if $('#discharged').attr('checked') is 'checked'
97     type = 'discharged'
98   else if $('#outpatient').attr('checked') is 'checked'
99     type = 'outpatient'
100   else
101     type = 'resident'
102     facilityNo = $('#facility').val()
103     roomNo = $('#room').val()
104     bedNo = $('#bed').val()
105
106
107   newPatient.fName = fName
108   newPatient.lName = lName
109   newPatient.patientID = patientID
110   newPatient.admittingPhysician = admittingPhysician
111
112   newPatient.address = address
113   newPatient.city = city
114   newPatient.state = state
115   newPatient.zip = zip
116   newPatient.phone = phone
117
118   newPatient.type = type
119   newPatient.facilityNo = facilityNo
120   newPatient.roomNo = roomNo
121   newPatient.bedNo = bedNo
122
123   console.log newPatient
124   # $.post("/newPatient", newPatient );
125
126   alert 'patient added'
```

127

128

129

```
1 // Generated by CoffeeScript 1.4.0
2 var facilities, findBeds, findFacilities, findRooms, hideResidentControls, initialize, pos
3
4 facilities = {};
5
6 $(document).ready(function() {
7   return initialize();
8 });
9
10 initialize = function() {
11   $('#residentControls').hide();
12   $('#discharged').click(hideResidentControls);
13   $('#outpatient').click(hideResidentControls);
14   $('#resident').click(showResidentControls);
15   $('#facility').change(function() {
16     return findRooms();
17   });
18   $('#room').change(function() {
19     return findBeds();
20   });
21   return findFacilities();
22 };
23
24 hideResidentControls = function() {
25   $('#residentControls').hide();
26   $('#room').html('');
27   $('#room').append('<option value="0">----</option>');
28   $('#bed').html('');
29   return $('#bed').append('<option value="0">----</option>');
30 };
31
32 showResidentControls = function() {
33   $('#residentControls').show();
34   return $('#facility').val(0);
35 };
36
37 findFacilities = function() {
38   var key, value;
39   for (key in residency) {
40     value = residency[key];
41     facilities[value.facilityNo] = value.facilityName;
42   }
43   for (key in facilities) {
44     value = facilities[key];
45     $('#facility').append('<option value="' + key + '">' + value + '</option>');
46   }
47   return findRooms();
48 };
49
50 findRooms = function() {
51   var key, selectedOption, tempRooms, value;
52   $('#room').html('');
53   tempRooms = {};
54   selectedOption = parseInt($('#facility option:selected').attr('value'));
55   for (key in residency) {
56     value = residency[key];
57     if (value.facilityNo === selectedOption) {
58       tempRooms[value.roomNo] = value.roomNo;
59     }
60   }
61   for (key in tempRooms) {
62     value = tempRooms[key];
63     $('#room').append('<option value="' + key + '">' + value + '</option>');
```

```
64     }
65     return findBeds();
66 };
67
68 findBeds = function() {
69     var key, selectedOption, tempBeds, value, _results;
70     $('#bed').html('');
71     tempBeds = {};
72     selectedOption = parseInt($('#room option:selected').attr('value'));
73     for (key in residency) {
74         value = residency[key];
75         if (value.roomNo === selectedOption) {
76             tempBeds[value.bedNo] = value.bedNo;
77         }
78     }
79     _results = [];
80     for (key in tempBeds) {
81         value = tempBeds[key];
82         _results.push($('#bed').append('<option value="' + key + '">' + value + '</option>'));
83     }
84     return _results;
85 };
86
87 postNewPatient = function() {
88     var address, admittingPhysician, bedNo, city, fName, facilityNo, lName, newPatient, pati
89     newPatient = {};
90     fName = $('#inputFName').val();
91     lName = $('#inputLName').val();
92     patientID = '54902';
93     admittingPhysician = '439023';
94     address = $('#inputAddress').val();
95     city = $('#inputCity').val();
96     state = $('#inputState').val();
97     zip = $('#inputZipcode').val();
98     phone = $('#inputPhone').val();
99     type = null;
100     facilityNo = null;
101     roomNo = null;
102     bedNo = null;
103     if ($('#discharged').attr('checked') === 'checked') {
104         type = 'discharged';
105     } else if ($('#outpatient').attr('checked') === 'checked') {
106         type = 'outpatient';
107     } else {
108         type = 'resident';
109         facilityNo = $('#facility').val();
110         roomNo = $('#room').val();
111         bedNo = $('#bed').val();
112     }
113     newPatient.fName = fName;
114     newPatient.lName = lName;
115     newPatient.patientID = patientID;
116     newPatient.admittingPhysician = admittingPhysician;
117     newPatient.address = address;
118     newPatient.city = city;
119     newPatient.state = state;
120     newPatient.zip = zip;
121     newPatient.phone = phone;
122     newPatient.type = type;
123     newPatient.facilityNo = facilityNo;
124     newPatient.roomNo = roomNo;
125     newPatient.bedNo = bedNo;
126     console.log(newPatient);
```

```
127 |   return alert('patient added');  
128 | };  
129 |
```

```
1  /* =====
2  * bootstrap-alert.js v2.3.1
3  * http://twitter.github.com/bootstrap/javascript.html#alerts
4  * =====
5  * Copyright 2012 Twitter, Inc.
6  *
7  * Licensed under the Apache License, Version 2.0 (the "License");
8  * you may not use this file except in compliance with the License.
9  * You may obtain a copy of the License at
10 *
11 * http://www.apache.org/licenses/LICENSE-2.0
12 *
13 * Unless required by applicable law or agreed to in writing, software
14 * distributed under the License is distributed on an "AS IS" BASIS,
15 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
16 * See the License for the specific language governing permissions and
17 * limitations under the License.
18 * ===== */
19
20
21 !function ($) {
22
23   "use strict"; // jshint ;_;
24
25   /* ALERT CLASS DEFINITION
26    * ===== */
27
28   var dismiss = '[data-dismiss="alert"]'
29   , Alert = function (el) {
30     $(el).on('click', dismiss, this.close)
31   }
32
33   Alert.prototype.close = function (e) {
34     var $this = $(this)
35     , selector = $this.attr('data-target')
36     , $parent
37
38     if (!selector) {
39       selector = $this.attr('href')
40       selector = selector && selector.replace(/.*(?=#[^\s]*$)/, '') //strip for ie7
41     }
42
43     $parent = $(selector)
44
45     e && e.preventDefault()
46
47     $parent.length || ($parent = $this.hasClass('alert') ? $this : $this.parent())
48
49     $parent.trigger(e = $.Event('close'))
50
51     if (e.isDefaultPrevented()) return
52
53     $parent.removeClass('in')
54
55     function removeElement() {
56       $parent
57         .trigger('closed')
58         .remove()
59     }
60
61     $.support.transition && $parent.hasClass('fade') ?
62       $parent.on($.support.transition.end, removeElement) :
```



```
64     removeElement()
65 }
66
67
68 /* ALERT PLUGIN DEFINITION
69 * ===== */
70
71 var old = $.fn.alert
72
73 $.fn.alert = function (option) {
74     return this.each(function () {
75         var $this = $(this)
76         , data = $this.data('alert')
77         if (!data) $this.data('alert', (data = new Alert(this)))
78         if (typeof option == 'string') data[option].call($this)
79     })
80 }
81
82 $.fn.alert.Constructor = Alert
83
84
85 /* ALERT NO CONFLICT
86 * ===== */
87
88 $.fn.alert.noConflict = function () {
89     $.fn.alert = old
90     return this
91 }
92
93
94 /* ALERT DATA-API
95 * ===== */
96
97 $(document).on('click.alert.data-api', dismiss, Alert.prototype.close)
98
99 }(window.jQuery);
```

```
1 /**
2  * Bootstrap.js by @fat & @mdo
3  * plugins: bootstrap-alert.js
4  * Copyright 2012 Twitter, Inc.
5  * http://www.apache.org/licenses/LICENSE-2.0.txt
6  */
7  !function(a){var
... b='[data-dismiss="alert"]',c=function(c){a(c).on("click",b,this.close)};c.prototype.close=
... function(b){function f(){e.trigger("closed").remove()}var
... c=a(this),d=c.attr("data-target"),e;d||(d=c.attr("href"),d=d&&d.replace(/.*(?=#[^\s]*$)/,""
... )),e=a(d),b&&b.preventDefault(),e.length||(e=c.hasClass("alert"?c.parent():e.trigger(b=
... a.Event("close")));if(b.isDefaultPrevented())return;e.removeClass("in"),a.support.transition
... &&e.hasClass("fade"?e.on(a.support.transition.end,f):f());var
... d=a.fn.alert;a.fn.alert=function(b){return this.each(function(){var
... d=a(this),e=d.data("alert");e||d.data("alert",e=new c(this)),typeof
... b=="string"&&e[b].call(d)}}},a.fn.alert.Constructor=c,a.fn.alert.noConflict=function(){
... return
... a.fn.alert=d,this},a(document).on("click.alert.data-api",b,c.prototype.close)}(window.
... jQuery)
```

```
1 #dear future self,
2 # you're creating json that matches the search field by using regex expressions in the .ma
3
4 # search json for data
5 # create new json list with data
6 # populate table with new json
7 patientID = 1 #cong -change
8 selectedDiagnosis = null;
9 input = null;
10 iNumber = null;
11 iString = '';
12 currentPatient = null
13 selectedDiagnosis = null
14 # patients initialized in PatientJSON.js
15
16 $(document).ready ->
17   do initialize
18
19
20 initialize = ->
21   # do getPatientNo
22   $('#searchBox').bind('propertychange input paste', findText)
23   do populateTable
24   do populatePatientInfo
25
26
27 populatePatientInfo = ->
28   $('#name').html(currentPatient.fname + ' ' + currentPatient.lname)
29   $('#phone').html(currentPatient.phone)
30   $('#address').html(currentPatient.address)
31   $('#cityStateZip').html(currentPatient.city + ', ' + currentPatient.state + ' ' + curr
32   $('#patientId').html(currentPatient.patientId)
33   $('#referringPhysician').html(currentPatient.referringPhysician)
34   $('#admissionType').html(currentPatient.patientType)
35
36
37
38 populateTable = ->
39   for key, value of diagnosis
40     name = value.name
41     diagnosisID = value.id
42
43     $('#dataTableBody').append('
44       <tr id="'+diagnosisID+'" onclick=selectDiagnosis('+ diagnosisID + ') >
45         <td>'+name+'</td>
46         <td>'+diagnosisID+'</td>
47       </tr>')
48
49 findText = ->
50   input = $('#searchBox').val()
51   iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase()
52   iString = input.replace(/[0-9]/g, '').toLowerCase()
53   do filterTable
54
55
56 filterTable = ->
57   if input == null or input == ''
58     # console.log 'input empty'
59     $('#dataTableBody tr').show()
60   else
61
62     if iString == null or iString == ''
63       regexString = null
```

```
64     else
65         regexString = new RegExp(iString, 'g' )
66
67
68     if iNumber == null or iNumber == ''
69         regexNumber = null
70     else
71         regexNumber = new RegExp(iNumber, 'g' )
72
73     # #filter patients
74     for key, value of diagnosis
75         name = value.name.toLowerCase()
76
77         console.log '--- ---'
78         if regexNumber != null and key.match(regexNumber)
79             console.log 'match found in key:' + key
80             $('#'+key).show()
81
82         else if regexString != null and name.match(regexString)
83             console.log 'match found in string: ' + value.fname + ' ' + value.lname
84             $('#'+key).show()
85         else
86             console.log 'match not found. hiding key ' + key
87             $('#'+key).hide()
88
89     selectDiagnosis = (diagnosisID)->
90         console.log ' key isnt modified yet'
91         selectedDiagnosis = diagnosisID
92         $('#diagnoseButton').attr('disabled', false)
93         $('tr').removeClass('highlight')
94         $('#'+diagnosisID).addClass('highlight')
95
96
97
98     diagnosePatient = ->
99         postVar = {}
100         postVar.patientID=currentPatient
101         postVar.diagnosisID = selectedDiagnosis
102
103         $.post("/pageToPostTo", postVar );
104         alert('patient diagnosed')
105
106         # now populate the table with all of that data
107         # $("#dataTable").find("tr:gt(0)").remove();
108
109         # Hide all rows but the ones found
110
111
```

```
1 // Generated by CoffeeScript 1.4.0
2 var currentPatient, diagnosePatient, filterTable, findText, iNumber, iString, initialize,
... input, patientID, populatePatientInfo, populateTable, selectDiagnosis, selectedDiagnosis;
3
4 patientID = 1;
5
6 selectedDiagnosis = null;
7
8 input = null;
9
10 iNumber = null;
11
12 iString = '';
13
14 currentPatient = null;
15
16 selectedDiagnosis = null;
17
18 $(document).ready(function() {
19   return initialize();
20 });
21
22 initialize = function() {
23   $('#searchBox').bind('propertychange input paste', findText);
24   populateTable();
25   return populatePatientInfo();
26 };
27
28 populatePatientInfo = function() {
29   $('#name').html(currentPatient.fname + ' ' + currentPatient.lname);
30   $('#phone').html(currentPatient.phone);
31   $('#address').html(currentPatient.address);
32   $('#cityStateZip').html(currentPatient.city + ', ' + currentPatient.state + ' ' +
... currentPatient.zip);
33   $('#patientId').html(currentPatient.patientId);
34   $('#referringPhysician').html(currentPatient.referringPhysician);
35   return $('#admissionType').html(currentPatient.patientType);
36 };
37
38 populateTable = function() {
39   var diagnosisID, key, name, value, _results;
40   _results = [];
41   for (key in diagnosis) {
42     value = diagnosis[key];
43     name = value.name;
44     diagnosisID = value.id;
45     _results.push($('#dataTableBody').append('\
46       <tr id="' + diagnosisID + '" onclick=selectDiagnosis(' + diagnosisID + ') >\
47         <td>' + name + '</td>\
48         <td>' + diagnosisID + '</td>\
49       </tr>'));
50   }
51   return _results;
52 };
53
54 findText = function() {
55   input = $('#searchBox').val();
56   iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase();
57   iString = input.replace(/[0-9]/g, '').toLowerCase();
58   return filterTable();
59 };
60
61 filterTable = function() {
```

```
62 var key, name, regexNumber, regexString, value, _results;
63 if (input === null || input === '') {
64     return $('#dataTableBody tr').show();
65 } else {
66     if (iString === null || iString === '') {
67         regexString = null;
68     } else {
69         regexString = new RegExp(iString, 'g');
70     }
71     if (iNumber === null || iNumber === '') {
72         regexNumber = null;
73     } else {
74         regexNumber = new RegExp(iNumber, 'g');
75     }
76     _results = [];
77     for (key in diagnosis) {
78         value = diagnosis[key];
79         name = value.name.toLowerCase();
80         console.log('----');
81         if (regexNumber !== null && key.match(regexNumber)) {
82             console.log('match found in key:' + key);
83             _results.push($('#' + key).show());
84         } else if (regexString !== null && name.match(regexString)) {
85             console.log('match found in string: ' + value.fname + ' ' + value.lname);
86             _results.push($('#' + key).show());
87         } else {
88             console.log('match not found. hiding key ' + key);
89             _results.push($('#' + key).hide());
90         }
91     }
92     return _results;
93 }
94 };
95
96 selectDiagnosis = function(diagnosisID) {
97     console.log(' key isnt modified yet');
98     selectedDiagnosis = diagnosisID;
99     $('#diagnoseButton').attr('disabled', false);
100     $('tr').removeClass('highlight');
101     return $('#' + diagnosisID).addClass('highlight');
102 };
103
104 diagnosePatient = function() {
105     var postVar;
106     postVar = {};
107     postVar.patientID = currentPatient;
108     postVar.diagnosisID = selectedDiagnosis;
109     $.post("/pageToPostTo", postVar);
110     return alert('patient diagnosed');
111 };
112
```

```
1 # post update data to website for patient stauts:
2 #http://coffeescriptcookbook.com/chapters/jquery/ajax
3
4 # get data from here: patient/patientId/
5 # update data post to here: patient/update/{id}
6
7 # show() resident patient information when resident patient is selected.
8
9
10 # working on getting patient information to display on click.
11
12
13
14 input = null;
15 iNumber = null;
16 iString = '';
17 currentPatientKey = null;
18 facilities = {}
19 # patients
20 storedInfo = {}
21
22
23
24
25 # patients initialized in PatientJSON.js
26
27 $(document).ready ->
28   do initialize
29     # $.getJSON('patientId') ->
30     #   patients = data
31     #   do initialize
32
33 initialize = ->
34   # patients = $.get 'testGet/patientID'
35   # this is disallowed in chrome, but will work if there is a local server running.
36   # It's how I will get the patientID JSON.
37   # right now, I am getting patients from a patientID.js file I created locally
38
39
40 $('#searchBox').bind('propertychange input paste', findText)
41 $('#residentControls').hide()
42 $('#discharged').click(hideResidentControls)
43 $('#outpatient').click(hideResidentControls)
44 $('#resident').click(showResidentControls)
45
46 do populateTable
47
48 $('#facility').change ->
49   do findRooms
50
51 $('#room').change ->
52   do findBeds
53
54 do findFacilities
55
56 populateTable = ->
57   for key, value of patients
58     fname = value.fname
59     lname = value.lname
60     dateAdmitted = value.dateAdmitted
61
62
```

```
63     $('#dataTableBody').append('
64     <tr id="'+value.patientId+'" onclick=displayInfo('+value.patientId+ ', '+
... key+')>
65         <td>'+fname+'</td>
66         <td>'+lname+'</td>
67         <td>'+dateAdmitted+'</td>
68         <td>'+value.patientId+'</td>
69     </tr>')
70
71
72 hideResidentControls = ->
73     $('#residentControls').hide()
74
75     $('#room').html('');
76     $('#room').append('<option value="0">----</option>')
77
78     $('#bed').html('');
79     $('#bed').append('<option value="0">----</option>')
80
81 showResidentControls = ->
82     $('#residentControls').show()
83     $('#facility').val(0) #default value is zero
84
85 findFacilities = ->
86     console.log 'findFacilities'
87     for key, value of residency
88         facilities[value.facilityNo] = value.facilityName
89
90     for key, value of facilities
91         $('#facility').append('<option value="'+key+'">'+value+'</option>')
92
93     do findRooms
94
95
96 findRooms = ->
97     console.log 'findRooms'
98     $('#room').html('');
99     tempRooms = {}
100
101     selectedOption = parseInt($('#facility option:selected').attr('value'))
102     for key, value of residency
103         if value.facilityNo is selectedOption
104             tempRooms[value.roomNo]=value.roomNo
105             #hacky. using key-value so that value will overwrite duplicates automatically
106
107     for key, value of tempRooms
108         $('#room').append('<option value="'+key+'">'+value+'</option>')
109
110     do findBeds
111
112 findBeds = ->
113     console.log 'findBeds'
114     $('#bed').html('');
115     tempBeds = {}
116
117     selectedOption = parseInt($('#room option:selected').attr('value'))
118
119     for key, value of residency
120         if value.roomNo is selectedOption
121             tempBeds[value.bedNo]=value.bedNo
122             #hacky. using key-value so that value will overwrite duplicates automatically
123
124     for key, value of tempBeds
```



```
125         $('#bed').append('<option value="'+key+'">'+value+'</option>')
126
127
128
129 findText = ->
130     input = $('#searchBox').val()
131     iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase()
132     iString = input.replace(/[0-9]/g, '').toLowerCase()
133     do filterTable
134
135 displayInfo = (patientID, key)->
136     currentPatientKey = patientID
137
138     $('#diagnoseButton').attr('disabled', false)
139     $('#deleteButton').attr('disabled', false)
140
141
142     $('#tr').removeClass('highlight')
143     $('#'+patientID).addClass('highlight')
144
145     patient = patients[key]
146     $('#name').val(patient.fname + ' ' + patient.lname)
147     $('#phone').val(patient.phone)
148     $('#address').val(patient.address)
149     $('#city').val(patient.city)
150     $('#zip').val(patient.zip)
151     $('#state').val(patient.state) #need to select state from list
152
153     $('#patientID').patientID
154     $('#referringPhysician').val(patient.referringPhysician)
155
156     if patient.patientType is 'Outpatient'
157         $('#outpatient').prop('checked', true)
158         do hideResidentControls
159
160     else if patient.patientType is 'Discharged'
161         $('#discharged').prop('checked', true)
162         # Remove all room and bed options
163         do hideResidentControls
164
165     else #residentPatient
166         $('#residentPatient').prop('Checked', true)
167         $('#residentControls').show()
168
169         # do findFacilities
170         $('#facility').val(patient.facilityNo)
171
172         do findRooms
173         $('#room').val(patient.roomNo)
174
175         do findBeds
176         $('#bed').val(patient.bedNo)
177
178         # $('#room').val(patient.roomNo)
179         # $('#bed').val(patient.bedNo)
180
181
182
183     # $('#patientType') populate options
184     # $('#careCenter') select from dropdown
185     # room select from dropdown
186     # bed select from dropdown
187
```

```
188
189 filterTable = ->
190   if input == null or input == ''
191     # console.log 'input empty'
192     $('#dataTableBody tr').show()
193   else
194
195     if iString == null or iString == ''
196       regexString = null
197     else
198       regexString = new RegExp(iString, 'g' )
199
200
201     if iNumber == null or iNumber == ''
202       regexNumber = null
203     else
204       regexNumber = new RegExp(iNumber, 'g' )
205
206     # #filter patients
207     for key, value of patients
208
209       if value.fname is not null
210         fname = value.fname.toLowerCase()
211       if value.lname is not null
212         lname = value.lname.toLowerCase()
213
214       console.log '---- ----'
215       if regexNumber != null and value.patientNo.match(regexNumber)
216         # console.log 'match found in key:' + key
217         $('#'+value.patientID).show()
218
219       else if regexString != null and (fname.match(regexString) or
220 ... lname.match(regexString))
221         # console.log 'match found in string: ' + value.fname + ' ' + value.lname
222         $('#'+value.patientID).show()
223       else
224         # console.log 'match not found. hiding key ' + key
225         $('#'+value.patientID).hide()
226
227     # now populate the table with all of that data
228     # $("#dataTable").find("tr:gt(0)").remove();
229
230     # Hide all rows but the ones found
231
232 postUpdate = ->
233   #currentPatientKey
234   patientUpdate = {} #this is the JSON object to be posted
235   pPhone = $('#phone').val()
236   pAddress = $('#address').val()
237   pCity = $('#city').val()
238   pState = $('#state').val()
239   pZip = $('#zip').val()
240   pType = null
241   pFacilityNo = 0
242   pRoomNo = 0
243   pBedNo = 0
244
245   if $('#discharged').attr('checked') is 'checked'
246     pType = 'Discharged'
247   else if $('#resident').attr('checked') is 'checked'
248     pType = 'Resident'
249   else
250     pType = 'Outpatient'
```

```
250
251   if pType is 'Resident'
252     pFacilityNo = $('#facility').val()
253     pRoomNo = $('#room').val()
254     pBedNo = $('#bed').val()
255
256   patientUpdate.patientNo = currentPatientKey
257   patientUpdate.phone = pPhone
258   patientUpdate.address= pAddress
259   patientUpdate.city = pCity
260   patientUpdate.state = pState
261   patientUpdate.zip = pZip
262   patientUpdate.pType = pType
263   patientUpdate.pFacilityNo = pFacilityNo
264   patientUpdate.pRoomNo = pRoomNo
265   patientUpdate.pBedNo = pBedNo
266
267
268   # will work on a server, but not on local filesystem. It's a security feature chrome
... has to prevent XSS
269
270   $.post("/pageToPostTo", patientUpdate );
271
272
273
274   # $.getJSON('patientId')->(data)
275   #   patients = data
276   #   do initialize
277
278   storedInfo.patientID = currentPatientKey
279   $('#'.currentPatientKey).click()
280   alert('patient updated')
281
282
283
284   deletePatient = ->
285     patientUpdate = {}
286     patientUpdate.patientNo = currentPatientKey
287     $.post("/pageToPostTo", patientUpdate );
288
289
```

```
1 // Generated by CoffeeScript 1.4.0
2 var currentPatientKey, deletePatient, displayInfo, facilities, filterTable, findBeds,
... findFacilities, findRooms, findText, hideResidentControls, iNumber, iString, initialize,
... input, populateTable, postUpdate, showResidentControls, storedInfo;
3
4 input = null;
5
6 iNumber = null;
7
8 iString = '';
9
10 currentPatientKey = null;
11
12 facilities = {};
13
14 storedInfo = {};
15
16 $(document).ready(function() {
17   return initialize();
18 });
19
20 initialize = function() {
21   $('#searchBox').bind('propertychange input paste', findText);
22   $('#residentControls').hide();
23   $('#discharged').click(hideResidentControls);
24   $('#outpatient').click(hideResidentControls);
25   $('#resident').click(showResidentControls);
26   populateTable();
27   $('#facility').change(function() {
28     return findRooms();
29   });
30   $('#room').change(function() {
31     return findBeds();
32   });
33   return findFacilities();
34 };
35
36 populateTable = function() {
37   var dateAdmitted, fname, key, lname, value, _results;
38   _results = [];
39   for (key in patients) {
40     value = patients[key];
41     fname = value.fname;
42     lname = value.lname;
43     dateAdmitted = value.dateAdmitted;
44     _results.push($('#dataTableBody').append('\
45       <tr id="' + value.patientId + '" onclick=displayInfo(' + value.patientId + ',' +
... + key + ')>\
46         <td>' + fname + '</td>\
47         <td>' + lname + '</td>\
48         <td>' + dateAdmitted + '</td>\
49         <td>' + value.patientId + '</td>\
50       </tr>'));
51   }
52   return _results;
53 };
54
55 hideResidentControls = function() {
56   $('#residentControls').hide();
57   $('#room').html('');
58   $('#room').append('<option value="0">----</option>');
59   $('#bed').html('');
60   return $('#bed').append('<option value="0">----</option>');
```

```
61 };
62
63 showResidentControls = function() {
64     $('#residentControls').show();
65     return $('#facility').val(0);
66 };
67
68 findFacilities = function() {
69     var key, value;
70     console.log('findFacilities');
71     for (key in residency) {
72         value = residency[key];
73         facilities[value.facilityNo] = value.facilityName;
74     }
75     for (key in facilities) {
76         value = facilities[key];
77         $('#facility').append('<option value="' + key + '">' + value + '</option>');
78     }
79     return findRooms();
80 };
81
82 findRooms = function() {
83     var key, selectedOption, tempRooms, value;
84     console.log('findRooms');
85     $('#room').html('');
86     tempRooms = {};
87     selectedOption = parseInt($('#facility option:selected').attr('value'));
88     for (key in residency) {
89         value = residency[key];
90         if (value.facilityNo === selectedOption) {
91             tempRooms[value.roomNo] = value.roomNo;
92         }
93     }
94     for (key in tempRooms) {
95         value = tempRooms[key];
96         $('#room').append('<option value="' + key + '">' + value + '</option>');
97     }
98     return findBeds();
99 };
100
101 findBeds = function() {
102     var key, selectedOption, tempBeds, value, _results;
103     console.log('findBeds');
104     $('#bed').html('');
105     tempBeds = {};
106     selectedOption = parseInt($('#room option:selected').attr('value'));
107     for (key in residency) {
108         value = residency[key];
109         if (value.roomNo === selectedOption) {
110             tempBeds[value.bedNo] = value.bedNo;
111         }
112     }
113     _results = [];
114     for (key in tempBeds) {
115         value = tempBeds[key];
116         _results.push($('#bed').append('<option value="' + key + '">' + value + '</option>'));
117     }
118     return _results;
119 };
120
121 findText = function() {
122     input = $('#searchBox').val();
123     iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase();
```

```
124 iString = input.replace(/[0-9]/g, '').toLowerCase();
125 return filterTable();
126 };
127
128 displayInfo = function(patientID, key) {
129     var patient;
130     currentPatientKey = patientID;
131     $('#diagnoseButton').attr('disabled', false);
132     $('#deleteButton').attr('disabled', false);
133     $('tr').removeClass('highlight');
134     $('# ' + patientID).addClass('highlight');
135     patient = patients[key];
136     $('#name').val(patient.fname + ' ' + patient.lname);
137     $('#phone').val(patient.phone);
138     $('#address').val(patient.address);
139     $('#city').val(patient.city);
140     $('#zip').val(patient.zip);
141     $('#state').val(patient.state);
142     $('#patientID').val(patientID);
143     $('#referringPhysician').val(patient.referringPhysician);
144     if (patient.patientType === 'Outpatient') {
145         $('#outpatient').prop('checked', true);
146         return hideResidentControls();
147     } else if (patient.patientType === 'Discharged') {
148         $('#discharged').prop('checked', true);
149         return hideResidentControls();
150     } else {
151         $('#residentPatient').prop('checked', true);
152         $('#residentControls').show();
153         $('#facility').val(patient.facilityNo);
154         findRooms();
155         $('#room').val(patient.roomNo);
156         findBeds();
157         return $('#bed').val(patient.bedNo);
158     }
159 };
160
161 filterTable = function() {
162     var fname, key, lname, regexNumber, regexString, value, _results;
163     if (input === null || input === '') {
164         return $('#dataTableBody tr').show();
165     } else {
166         if (iString === null || iString === '') {
167             regexString = null;
168         } else {
169             regexString = new RegExp(iString, 'g');
170         }
171         if (iNumber === null || iNumber === '') {
172             regexNumber = null;
173         } else {
174             regexNumber = new RegExp(iNumber, 'g');
175         }
176         _results = [];
177         for (key in patients) {
178             value = patients[key];
179             if (value.fname !== null) {
180                 fname = value.fname.toLowerCase();
181             }
182             if (value.lname !== null) {
183                 lname = value.lname.toLowerCase();
184             }
185             console.log('----');
186             if (regexNumber !== null && value.patientNo.match(regexNumber)) {
```

```
187     _results.push($('#' + value.patientID).show());
188   } else if (regexString !== null && (fname.match(regexString) ||
... lname.match(regexString))) {
189     _results.push($('#' + value.patientID).show());
190   } else {
191     _results.push($('#' + value.patientID).hide());
192   }
193 }
194 return _results;
195 }
196 };
197
198 postUpdate = function() {
199   var pAddress, pBedNo, pCity, pFacilityNo, pPhone, pRoomNo, pState, pType, pZip,
... patientUpdate;
200   patientUpdate = {};
201   pPhone = $('#phone').val();
202   pAddress = $('#address').val();
203   pCity = $('#city').val();
204   pState = $('#state').val();
205   pZip = $('#zip').val();
206   pType = null;
207   pFacilityNo = 0;
208   pRoomNo = 0;
209   pBedNo = 0;
210   if ($('#discharged').attr('checked') === 'checked') {
211     pType = 'Discharged';
212   } else if ($('#resident').attr('checked') === 'checked') {
213     pType = 'Resident';
214   } else {
215     pType = 'Outpatient';
216   }
217   if (pType === 'Resident') {
218     pFacilityNo = $('#facility').val();
219     pRoomNo = $('#room').val();
220     pBedNo = $('#bed').val();
221   }
222   patientUpdate.patientNo = currentPatientKey;
223   patientUpdate.phone = pPhone;
224   patientUpdate.address = pAddress;
225   patientUpdate.city = pCity;
226   patientUpdate.state = pState;
227   patientUpdate.zip = pZip;
228   patientUpdate.pType = pType;
229   patientUpdate.pFacilityNo = pFacilityNo;
230   patientUpdate.pRoomNo = pRoomNo;
231   patientUpdate.pBedNo = pBedNo;
232   $.post("/pageToPostTo", patientUpdate);
233   storedInfo.patientID = currentPatientKey;
234   $('#'.currentPatientKey).click();
235   return alert('patient updated');
236 };
237
238 deletePatient = function() {
239   var patientUpdate;
240   patientUpdate = {};
241   patientUpdate.patientNo = currentPatientKey;
242   return $.post("/pageToPostTo", patientUpdate);
243 };
244
```

AddPatient.js

```
1 // Generated by CoffeeScript 1.4.0
2 var facilities, findBeds, findFacilities, findRooms,
    hideResidentControls, initialize, postNewPatient,
    showResidentControls;
3
4 facilities = {};
5
6 $(document).ready(function() {
7     $.getJSON('count',function(data) {
8         $('#inputPatientID').val(data);
9     });
10
11     return initialize();
12 });
13
14 initialize = function() {
15     $('#residentControls').hide();
16     $('#discharged').click(hideResidentControls);
17     $('#outpatient').click(hideResidentControls);
18     $('#residentPatient').click(showResidentControls);
19     $('#inputAdmittingPhysician').val(currentPhysician);
20     $('#facility').change(function() {
21         return findRooms();
22     });
23     $('#room').change(function() {
24         return findBeds();
25     });
26     return findFacilities();
27 };
28
29 hideResidentControls = function() {
30     $('#residentControls').hide();
31     $('#room').html('');
32     $('#room').append('<option value="0">---</option>');
33     $('#bed').html('');
34     return $('#bed').append('<option value="0">---</option>');
35 };
36
37 showResidentControls = function() {
38     $('#residentControls').show();
39     return $('#facility').val(0);
```


AddPatient.js

```
40 };
41
42 findFacilities = function() {
43     var key, value;
44     for (key in residency) {
45         value = residency[key];
46         facilities[value.facilityNo] = value.facilityName;
47     }
48     for (key in facilities) {
49         value = facilities[key];
50         $('#facility').append('<option value="' + key + '">' + value +
            '</option>');
51     }
52     return findRooms();
53 };
54
55 findRooms = function() {
56     var key, selectedOption, tempRooms, value;
57     $('#room').html('');
58     tempRooms = {};
59     selectedOption = parseInt($('#facility
        option:selected').attr('value'));
60     for (key in residency) {
61         value = residency[key];
62         if (value.facilityNo === selectedOption) {
63             tempRooms[value.roomNo] = value.roomNo;
64         }
65     }
66     for (key in tempRooms) {
67         value = tempRooms[key];
68         $('#room').append('<option value="' + key + '">' + value + '</
            option>');
69     }
70     return findBeds();
71 };
72
73 findBeds = function() {
74     var key, selectedOption, tempBeds, value, _results;
75     $('#bed').html('');
76     tempBeds = {};
77     selectedOption = parseInt($('#room
```

AddPatient.js

```
option:selected').attr('value'));
78 for (key in residency) {
79     value = residency[key];
80     if (value.roomNo === selectedOption) {
81         tempBeds[value.bedNo] = value.bedNo;
82     }
83 }
84 _results = [];
85 for (key in tempBeds) {
86     value = tempBeds[key];
87     _results.push($('#bed').append('<option value="' + key + '">' +
    value + '</option>'));
88 }
89 return _results;
90};
91
92 postNewPatient = function() {
93     var address, admittingPhysician, bedNo, city, fName, facilityNo,
        lName, newPatient, patientID, phone, roomNo, state, type, zip;
94     newPatient = {};
95     fName = $('#inputFName').val();
96     lName = $('#inputLName').val();
97     patientID = $('#inputPatientID').val();
98     admittingPhysician = $('#inputAdmittingPhysician').val();
99     address = $('#inputAddress').val();
100    city = $('#inputCity').val();
101    state = $('#inputState').val();
102    zip = $('#inputZipcode').val();
103    phone = $('#inputPhone').val();
104    type = null;
105    facilityNo = 11;
106    roomNo = 1;
107    bedNo = 1;
108    if ($('#discharged').attr('checked') === 'checked') {
109        type = 'Discharged';
110    } else if ($('#residentPatient').attr('checked') === 'checked') {
111        type = 'Resident';
112    } else {
113        type = 'Outpatient';
114    }
115    if (type === 'Resident') {
```

AddPatient.js

```
116     facilityNo = $('#facility').val();
117     roomNo = $('#room').val();
118     bedNo = $('#bed').val();
119 }
120 newPatient.fName = fName;
121 newPatient.lName = lName;
122 newPatient.patientID = patientID;
123 newPatient.admittingPhysician = admittingPhysician;
124 newPatient.address = address;
125 newPatient.city = city;
126 newPatient.state = state;
127 newPatient.zip = zip;
128 newPatient.phone = phone;
129 newPatient.type = type;
130 newPatient.facilityNo = facilityNo;
131 newPatient.roomNo = roomNo;
132 newPatient.bedNo = bedNo;
133
134 $.post("create", newPatient);
135
136
137 return alert('patient added');
138 };
139
```

DiagnoseParser.js

```
1 // Generated by CoffeeScript 1.4.0
2 var currentPatient, diagnosePatient, filterTable, findText, iNumber,
  iString, initialize, input, patientID, populatePatientInfo,
  populateTable, selectDiagnosis, selectedDiagnosis;
3
4 patientID = 1;
5
6 selectedDiagnosis = null;
7
8 input = null;
9
10 iNumber = null;
11
12 iString = '';
13
14 currentPatient = null;
15
16 selectedDiagnosis = null;
17
18 $(document).ready(function() {
19   $.getJSON('diagnoseId/', function(data) {
20     diagnosis = data;
21   });
22
23   $.getJSON('../patient/patientId/'+currentPatientId, function(data) {
24     currentPatient = data;
25     return initialize();
26   });
27 });
28
29 initialize = function() {
30   $('#searchBox').bind('propertychange input paste', findText);
31   populateTable();
32   return populatePatientInfo();
33 };
34
35 populatePatientInfo = function() {
36   $('#name').html(currentPatient.fname + ' ' +
    currentPatient.lname);
37   $('#phone').html(currentPatient.phone);
38   $('#address').html(currentPatient.address);
```

DiagnoseParser.js

```
39     $('#cityStateZip').html(currentPatient.city + ', ' +
    currentPatient.state + ' ' + currentPatient.zip);
40     $('#patientId').html(currentPatient.patientId);
41     $
    ('#referringPhysician').html(currentPatient.referringPhysician);
42     return $('#admissionType').html(currentPatient.patientType);
43 };
44
45 populateTable = function() {
46     var diagnosisID, key, name, value, _results;
47     _results = [];
48     for (key in diagnosis) {
49         value = diagnosis[key];
50         name = value.name;
51         diagnosisID = value.id;
52         _results.push($('#dataTableBody').append('\
53             <tr id="' + diagnosisID + '" onclick=selectDiagnosis(' +
    diagnosisID + ') >\
54                 <td>' + name + '</td>\
55                 <td>' + diagnosisID + '</td>\
56             </tr>')));
57     }
58     return _results;
59 };
60
61 findText = function() {
62     input = $('#searchBox').val();
63     iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase();
64     iString = input.replace(/[0-9]/g, '').toLowerCase();
65     return filterTable();
66 };
67
68 filterTable = function() {
69     var key, name, regexNumber, regexString, value, _results;
70     if (input === null || input === '') {
71         return $('#dataTableBody tr').show();
72     } else {
73         if (iString === null || iString === '') {
74             regexString = null;
75         } else {
76             regexString = new RegExp(iString, 'g');
```

DiagnoseParser.js

```
77     }
78     if (iNumber === null || iNumber === '') {
79         regexNumber = null;
80     } else {
81         regexNumber = new RegExp(iNumber, 'g');
82     }
83     _results = [];
84     for (key in diagnosis) {
85         value = diagnosis[key];
86         name = value.name.toLowerCase();
87         console.log('--- ---');
88         if (regexNumber !== null && key.match(regexNumber)) {
89             console.log('match found in key:' + key);
90             _results.push($('#' + key).show());
91         } else if (regexString !== null && name.match(regexString)) {
92             console.log('match found in string: ' + value.fname + ' ' +
value.lname);
93             _results.push($('#' + key).show());
94         } else {
95             console.log('match not found. hiding key ' + key);
96             _results.push($('#' + key).hide());
97         }
98     }
99     return _results;
100 }
101 };
102
103 selectDiagnosis = function(diagnosisID) {
104     console.log(' key isnt modified yet');
105     selectedDiagnosis = diagnosisID;
106     $('#diagnoseButton').attr('disabled', false);
107     $('tr').removeClass('highlight');
108     return $('#' + diagnosisID).addClass('highlight');
109 };
110
111 diagnosePatient = function() {
112     var postVar;
113     postVar = {};
114     postVar.patientId = currentPatient.patientId;
115     postVar.diagnoseId = selectedDiagnosis;
116     postVar.physicianId = currentPhysician;
```

DiagnoseParser.js

```
117 $.post("../physicianDX/add", postVar);  
118 return alert('patient diagnosed');  
119 };  
120
```

OrderParser.js

```
1 // Generated by CoffeeScript 1.4.0
2 var filterTable, findText, iNumber, iString, initialize, input,
  populateTable;
3
4 input = null;
5
6 iNumber = null;
7
8 iString = '';
9
10 $(document).ready(function() {
11   return initialize();
12 });
13
14 initialize = function() {
15   $('#searchBox').bind('propertychange input paste', findText);
16   return populateTable();
17 };
18
19 populateTable = function() {
20   var key, name, value, _results;
21   _results = [];
22   for (key in treatments) {
23     value = treatments[key];
24     name = value.name;
25     _results.push($('#dataTableBody').append('\
26       <tr id="' + key + '">\
27         <td>' + name + '</td>\
28         <td>' + key + '</td>\
29       </tr>')));
30   }
31   return _results;
32 };
33
34 findText = function() {
35   input = $('#searchBox').val();
36   iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase();
37   iString = input.replace(/[0-9]/g, '').toLowerCase();
38   return filterTable();
39 };
40
```


OrderParser.js

```
41 filterTable = function() {
42   var key, name, regexNumber, regexString, value, _results;
43   if (input === null || input === '') {
44     return $('#dataTableBody tr').show();
45   } else {
46     if (iString === null || iString === '') {
47       regexString = null;
48     } else {
49       regexString = new RegExp(iString, 'g');
50     }
51     if (iNumber === null || iNumber === '') {
52       regexNumber = null;
53     } else {
54       regexNumber = new RegExp(iNumber, 'g');
55     }
56     _results = [];
57     for (key in treatments) {
58       value = treatments[key];
59       name = value.name.toLowerCase();
60       console.log('--- ---');
61       if (regexNumber !== null && key.match(regexNumber)) {
62         console.log('match found in key:' + key);
63         _results.push($('#' + key).show());
64       } else if (regexString !== null && name.match(regexString)) {
65         console.log('match found in string: ' + value.fname + ' ' +
66           value.lname);
67         _results.push($('#' + key).show());
68       } else {
69         console.log('match not found. hiding key ' + key);
70         _results.push($('#' + key).hide());
71       }
72     }
73     return _results;
74   }
75 }
```

PatientParser.js

```
1 // Generated by CoffeeScript 1.4.0
2 var currentPatientKey, displayInfo, facilities, filterTable,
  findBeds, findFacilities, findRooms, findText, hideResidentControls,
  iNumber, iString, initialize, input, patients, populateTable,
  postUpdate, showResidentControls;
3
4 input = null;
5
6 iNumber = null;
7
8 iString = '';
9
10 currentPatientKey = null;
11
12 facilities = {};
13
14 patients = {};
15
16 storedInfo = {};
17
18 diagnoseURL = null;
19
20 $(document).ready(function() {
21   return $.getJSON('patientId/',function(data) {
22     patients = data;
23     return initialize();
24   });
25 });
26
27 initialize = function() {
28   $('#searchBox').bind('propertychange input paste', findText);
29   $('#residentControls').hide();
30   $('#discharged').click(hideResidentControls);
31   $('#outpatient').click(hideResidentControls);
32   $('#residentPatient').click(showResidentControls);
33   populateTable();
34   $('#facility').change(function() {
35     return findRooms();
36   });
37   $('#room').change(function() {
38     return findBeds();
```

PatientParser.js

```
39 });
40 return findFacilities();
41 };
42
43 populateTable = function() {
44     var dateAdmitted, fname, key, lname, value, _results;
45     _results = [];
46     $("#dataTableBody tr").remove();
47     for (key in patients) {
48         value = patients[key];
49         fname = value.fname;
50         lname = value.lname;
51         patientId = value.patientId;
52         dateAdmitted = value.dateAdmitted;
53         _results.push($("#dataTableBody").append('\
54             <tr id="' + patientId + '" onclick=displayInfo(' +
patientId + ', ' + key + ')>\
55                 <td>' + fname + '</td>\
56                 <td>' + lname + '</td>\
57                 <td>' + dateAdmitted + '</td>\
58                 <td>' + patientId + '</td>\
59             </tr>')));
60     }
61     return _results;
62 };
63
64
65
66
67 hideResidentControls = function() {
68     $('#residentControls').hide();
69     $('#room').html('');
70     $('#room').append('<option value="0">---</option>');
71     $('#bed').html('');
72     return $('#bed').append('<option value="0">---</option>');
73 };
74
75 showResidentControls = function() {
76     $('#residentControls').show();
77     return $('#facility').val(0);
78 };
```

PatientParser.js

```
79
80 findFacilities = function() {
81   var key, value;
82   console.log('findFacilities');
83   for (key in residency) {
84     value = residency[key];
85     facilities[value.facilityNo] = value.facilityName;
86   }
87   for (key in facilities) {
88     value = facilities[key];
89     $('#facility').append('<option value="' + key + '">' + value +
    '</option>');
90   }
91   return findRooms();
92 };
93
94 findRooms = function() {
95   var key, selectedOption, tempRooms, value;
96   console.log('findRooms');
97   $('#room').html('');
98   tempRooms = {};
99   selectedOption = parseInt($('#facility
    option:selected').attr('value'));
100  for (key in residency) {
101    value = residency[key];
102    if (value.facilityNo === selectedOption) {
103      tempRooms[value.roomNo] = value.roomNo;
104    }
105  }
106  for (key in tempRooms) {
107    value = tempRooms[key];
108    $('#room').append('<option value="' + key + '">' + value + '</
    option>');
109  }
110  return findBeds();
111 };
112
113 findBeds = function() {
114   var key, selectedOption, tempBeds, value, _results;
115   console.log('findBeds');
116   $('#bed').html('');
```

PatientParser.js

```
117 tempBeds = {};  
118 selectedOption = parseInt($('#room  
option:selected').attr('value'));  
119 for (key in residency) {  
120     value = residency[key];  
121     if (value.roomNo === selectedOption) {  
122         tempBeds[value.bedNo] = value.bedNo;  
123     }  
124 }  
125 _results = [];  
126 for (key in tempBeds) {  
127     value = tempBeds[key];  
128     _results.push($('#bed').append('<option value="' + key + '">' +  
value + '</option>'));  
129 }  
130 return _results;  
131 };  
132  
133 findText = function() {  
134     input = $('#searchBox').val();  
135     iNumber = input.replace(/[a-zA-z]/g, '').toLowerCase();  
136     iString = input.replace(/[0-9]/g, '').toLowerCase();  
137     return filterTable();  
138 };  
139  
140 displayInfo = function(patientID, key) {  
141     var patient;  
142     $('#deleteButton').attr('disabled', false);  
143     $('#diagnose').attr("action", "../diagnose/?  
patientId="+patientID);  
144     currentPatientKey = patientID;  
145     $('#tr').removeClass('highlight');  
146     $('#'+ patientID).addClass('highlight');  
147     patient = patients[key];  
148     $('#name').val(patient.fname + ' ' + patient.lname);  
149     $('#phone').val(patient.phone);  
150     $('#address').val(patient.address);  
151     $('#city').val(patient.city);  
152     $('#zip').val(patient.zip);  
153     $('#state').val(patient.state);  
154     $('#patientID').val(patientID);
```

PatientParser.js

```
155 $('#referringPhysician').val(patient.referringPhysician);
156 if (patient.patientType === 'Outpatient') {
157     $('#outpatient').prop('checked', true);
158     return hideResidentControls();
159 } else if (patient.patientType === 'Discharged') {
160     $('#discharged').prop('checked', true);
161     return hideResidentControls();
162 } else {
163     $('#residentPatient').prop('checked', true);
164     $('#residentControls').show();
165     $('#facility').val(patient.facilityNo);
166     findRooms();
167     $('#room').val(patient.roomNo);
168     findBeds();
169     return $('#bed').val(patient.bedNo);
170 }
171 };
172
173 filterTable = function() {
174     var fname, key, lname, regexNumber, regexString, value, _results;
175     if (input === null || input === '') {
176         return $('#dataTableBody tr').show();
177     } else {
178         if (iString === null || iString === '') {
179             regexString = null;
180         } else {
181             regexString = new RegExp(iString, 'g');
182         }
183         if (iNumber === null || iNumber === '') {
184             regexNumber = null;
185         } else {
186             regexNumber = new RegExp(iNumber, 'g');
187         }
188         _results = [];
189         for (key in patients) {
190             value = patients[key];
191             fname = (fname === null)? null : value.fname.toLowerCase();
192             lname = (lname === null)? null : value.lname.toLowerCase();
193             console.log('--- ---');
194             if (regexNumber !== null &&
                value.patientId.toString().match(regexNumber)) {
```

PatientParser.js

```
195     _results.push($('#' + value.patientId).show());
196   } else if (regexString !== null && (fname.match(regexString)
197   || lname.match(regexString))) {
198     _results.push($('#' + value.patientId).show());
199   } else {
200     _results.push($('#' + value.patientId).hide());
201   }
202   return _results;
203 }
204 };
205
206 deletePatient = function() {
207   $.getJSON("delete/"+currentPatientKey,function(data){
208     patients = data;
209     initialize();
210   });
211
212
213 };
214
215 postUpdate = function() {
216   var pAddress, pBedNo, pCity, pFacility, pFacilityNo, pPhone,
217   pRoomNo, pState, pType, pZip, patientUpdate;
218   patientUpdate = {};
219   pPhone = $('#phone').val();
220   pAddress = $('#address').val();
221   pCity = $('#city').val();
222   pState = $('#state').val();
223   pZip = $('#zip').val();
224   pType = null;
225   pFacilityNo = 0;
226   pRoomNo = 0;
227   pBedNo = 0;
228   if ($('#discharged').attr('checked') === 'checked') {
229     pType = 'Discharged';
230   } else if ($('#residentPatient').attr('checked') === 'checked') {
231     pType = 'Resident';
232   } else {
233     pType = 'Outpatient';
234   }
```

PatientParser.js

```
234  if (pType === 'Resident') {
235      pFacilityNo = $('#facility').val();
236      pRoomNo = $('#room').val();
237      pBedNo = $('#bed').val();
238  }
239  patientUpdate.patientNo = currentPatientKey;
240  patientUpdate.phone = pPhone;
241  patientUpdate.address = pAddress;
242  patientUpdate.city = pCity;
243  patientUpdate.state = pState;
244  patientUpdate.zip = pZip;
245  patientUpdate.pType = pType;
246  patientUpdate.pFacilityNo = pFacilityNo;
247  patientUpdate.pRoomNo = pRoomNo;
248  patientUpdate.pBedNo = pBedNo;
249  $.post("update/"+patientUpdate.patientNo, patientUpdate);
250  $.getJSON('patientId/',function(data) {
251      patients = data;
252      return initialize();
253  });
254  storedInfo.patientID = currentPatientKey;
255  $('#'.currentPatientKey).click();
256  return alert('patient updated');
257 };
258
```