



Legend: Because there is not a linear time flow to events, since users have many options at the home page, and the ability to return home from any page, data flows are labelled and grouped by process.

The home page process is labelled 3.0. From the home page, an arrow labelled 1 is anything that a user can do from their home page, and subsequent numbers occur following the arrows from the first request, usually culminating in the user pressing the back button and being taken back home.

For example, at the home page, a user can choose to do anything labelled with a 1.

Once the user has chosen something, if that action requires more than one data handoff, they are labelled in order, following the arrows, from 2 on.

Every process labelled 4.0 also is displaying a back button that will take the user (and data flow) back to process 3.0

There is no room to add arrows from the 6.0 processes back to 3.0, so we omitted them. Assume they exist

Also note that I have duplicated some commonly used Data Stores to reduce clutter around them in the diagram.

The actual system will have one of each label of Data Store, with the exception of D3, which is the database itself, taking raw user-defined SQL as input

Function: Find a patient's list of medical records Inputs: @Patient_SIN Outputs: @RECORD.* Connect to Database Query = SELECT * FROM RECORDS WHERE @Patient_SIN = RECORDS.Patient SIN

Parse Query Execute Query

Close connection to Database

Function: View Institution calendar(Patient) Inputs: @Inst_Name

Outputs: @DaysClosed Pseudocode: Connect to Database

Query = **SELECT** DaysClosed **FROM** Days Closed, Med Institution WHERE DaysClosed.MedInst_ID = Med Institution.MedInst_ID AND Med Institution.InstName = @InstName

Parse Query **Execute Query** Close connection to Database Function: View Institution calendar(Doctor) Inputs: @Inst_Name Outputs: @DavsClosed Pseudocode: Connect to Database

Query = SELECT DaysClosed FROM Days Closed, Med Institution WHERE DaysClosed.MedInst_ID = Med Institution.MedInst_ID **AND** Med Institution.InstName = @InstName

Parse Query **Execute Query** Close connection to Database

Function: List Prescriptions(Patient) Inputs: @Patient_SIN Outputs: @presType, @presName, @length, @ Times_Renewable Pseudocode: Connect to Database

WHERE P.Patient_SIN = @Patient_SIN Parse Query **Execute Query**

Query = **SELECT** * **FROM** Prescriptions **AS** P;

Function: Find the work schedule and work location of a medical professional Inputs: @MedStaff_SIN Outputs: WORK_PLACEMENT.dayWorking, WORK_PLACEMENT.workLocationName Connect to Database Query = **SELECT** dayWorking, workLocationName FROM WORK_PLACEMENT AS P WHERE @MedStaff_SIN = P.MedStaff_SIN Parse Query

Pseudocode: Connect to Database Query = **INSERT INTO** Prescription **VALUES** (@presType, @presName, @duration, @renewableFlag, @ID); Parse Query Execute Query

Inputs: @presType, @presName, @duration, @renewableFlag, @ID

Function: Add Prescription(Prescription)

Outputs: None

Close connection to Database

Function: Check login details(Medical Staff) Inputs:@Username, @Password Outputs: @Fname, @Lname, @isAdmin Connect to Database

Query = **SELECT** Fname, Lname, isAdmin FROM Medical Staff

WHERE Medical Staff.Username = @Username, Medical Staff.Password = @Password

Parse Query **Execute Query** Close connection to Database Function: Check login details(Patient) Inputs:@Username, @Password Outputs: @Fname, @Lname, @isAdmin Pseudocode Connect to Database

Query = **SELECT** Fname, Lname, isAdmin **FROM** Patient

WHERE Patient. Username = @Username, Patient. Password = @Password

Parse Query **Execute Query** Close connection to Database Function: View Specific Table(Admin) Inputs: @TableName Outputs: @TableName.* Pseudocode: Connect to Database

Query = **SELECT** * FROM @TableName;

Parse Query **Execute Query** Close connection to Database Function: Add Record(Doctor) Inputs: @ID, @Date Outputs: None Pseudocode: Connect to Database

Close connection to Database

Query = INSERT INTO Records VALUES (@ID, @Date);

Parse Query Execute Query Close connection to Database

Function: Add Attachment(Doctor) Inputs: @docName, @Type, @Record_ID Outputs: None Pseudocode: Connect to Database Query = INSERT INTO Attachment VALUES (@Record_ID, @docName, @Type); Parse Query Execute Query Close connection to Database

and get all their information Inputs: @PATIENT.Fname, @PATIENT.Lname Outputs: PATIENT.* Pseudocode: Connect to Database Query = **SELECT** * **FROM** PATIENT **AS** P1 **WHERE** P1.FName = @FName AND P1.LName = @LName; Parse Query

Function: Search patients by first and last name,

Function: Get all medical history items for a given user (requires multiple tables) Inputs: @Patient SSN Outputs: Med_Hist_ID, Diagnosis, Recommended treatment, Notes

Pseudocode: Connect to Database

Query = SELECT Med_Hist_ID, Diagnosis, Recommended treatment, Notes FROM Record AS R, Medical History Item AS MHI WHERE R.Patient_SIN = @Patient_SIN

Parse Query **Execute Query** Close connection to Database

AND MHI.Record_ID = R.Record_ID

Function: Get complete list of users Inputs: none Outputs: users Pseudocode: Connect to Database

Query = SELECT FName, LName, isAdmin FROM Patients, Medical Staff Parse Query **Execute Query** Close connection to Database

Function: Get Future Appointments Inputs:@Patient SSN, @Current Date Outputs: @appointmentType, @Date, @Start_Time, @End_Time Pseudocode: Connect to Database

Query = **SELECT** appointmentType, Date, Start_Time, End_Time FROM Appointment WHERE Appointment.Patient_SIN = @Patient_SSN AND Appointment.Date >= @Current_Date

Parse Query **Execute Query**

Close connection to Database

Function: Get list of patients that given medical staff has access to Inputs:@MedStaff SIN

Outputs: Fname, Lname Pseudocode: Connect to Database

Query = **SELECT** Fname, Lname, FROM Patient as P

WHERE EXISTS (RECORD AS R AND ACCESS AS A AND P.patient_SIN = R.Patient_SIN AND A.MedStaff_SIN = @MedStaff_SIN AND A.RECORD_ID = R.Record_ID)

Execute Query

Close connection to Database

Parse Query Execute Query Close connection to Database

Inputs: None Outputs: @Fname, @Lname Pseudocode: Connect to Database

Function: Get FName, LName of all medical professionals

Query = **SELECT** Fname, Lname **FROM** Medical Staff

Parse Query Execute Query

Close connection to Database

Function: Get attachments and symptoms for a given medical history item Inputs: @Med_Hist_ID

Outputs: DocName, Type, Symptom Connect to Database

Query = **SELECT** DocName, Type FROM Attachment AS A, Medical History Item AS MHI WHERE A.Record_ID = MHI.Record_ID AND MHI.Med_Hist_ID = @Med_Hist_ID

Parse Query **Execute Query**

Query2 = **SELECT** Symptom FROM Med Hist Symptoms AS MHS, Medical History Item AS MHI WHERE MHS Med_Hist_ID= @Med_Hist_ID

Parse Query **Execute Query** Close connection to Database **Function:** Get all data for a given patient

Inputs: @patient_SIN

Outputs: MedHistoryItem.diagnosis, MedHistoryItem.recommendedTreatment, MedHistoryItem.notes

Prescription.presType, Prescription.presName, Prescription.length, Prescription.timesRenewable, MedHistSymptoms. symptom Pseudocode: Connect to Database

Query = **SELECT** MedHistoryItem.diagnosis, MedHistoryItem.recommendedTreatment, MedHistoryItem.notes, Record.Date, Prescription.presType, Prescription.presName, Prescription.length, Prescription.timesRenewable, MedHistSymptoms. symptom,

FROM MedHistoryItem, Prescription, MedHistSymptoms, Records, Insurance WHERE @patient_SIN = Record.patient_SIN AND Record.Record_ID = MedHistoryItem.Record_ID

AND MedHistSymptoms.medHist_ID = MedHistoryItem.MedHistItem_ID AND Prescription.Patient_SIN = @patient_SIN AND @patient_SIN = Insurance.Patient_SIN

Parse Query **Execute Query**

Close connection to Database

Function: Get list of all tables in the database Inputs: none Outputs: information_scheme.tables.table_name.* Pseudocode: Connect to Database

Query = SELECT table_name FROM information_schema.tables WHERE table_schema = 'Medical Database'; Parse Query **Execute Query** Close connection to Database

Function: Get appointments for a given medical professional Inputs:@MedStaff_SSN, Outputs: appointmentType, Date, Start_Time, End_Time Pseudocode Connect to Database

Execute Query

Close connection to Database

Query = **SELECT** appointmentType, Date, Start_Time, End_Time **FROM** Appointment WHERE Appointment.MedStaff_SSN= @MedStaff_SSN

Parse Query **Execute Query**

Close connection to Database

Function: Add appointment to appointments table, given all info @MedStaff_SSN, @appointmentType, @Date, @Start_Time, @End_Time Outputs: None Pseudocode: Connect to Database

Query = INSERT INTO Appointment VALUES (@Inst_Name, @Patient_SSN @MedStaff_SSN, @appointmentType, @Date, @Start_Time, @End_Time)

Parse Query Execute Query Close connection to Database