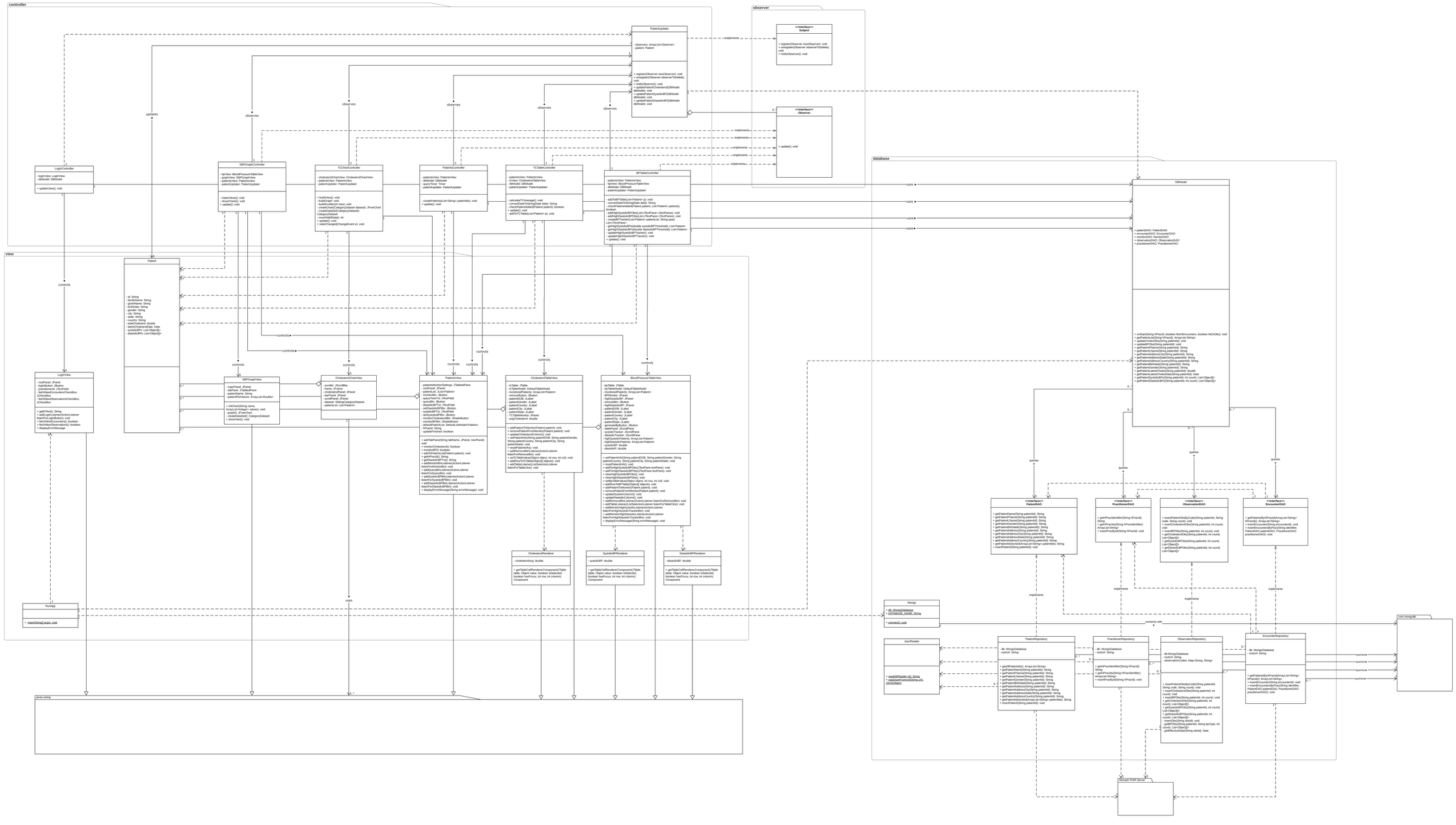
controller
observer
database
view
PatientUpdater
- observers: ArrayList<Observer>
- patient: Patient
+ register(Observer newObserver): void
+ unregister(Observer observerToDelete): void
+ notify(Observer): void
+ updatePatientCholesterolDB(Model dbModel): void
+ updatePatientSystolicBPDB(Model dbModel): void
+ updatePatientDiastolicBPDB(Model dbModel): void
<<interface>>
Subject
+ register(Observer newObserver): void
+ unregister(Observer observerToDelete): void
+ notify(Observer): void
<<interface>>
Observer
+ update(): void
LoginController
- loginView: LoginView
- dbModel: DBModel
+ updateView(): void
SBPGraphController
- bpView: BloodPressureTableView
- graphView: SBPGraphView
- patientView: PatientsView
- patientUpdater: PatientUpdater
- makeViews(): void
- showCharts(): void
+ update(): void
TCChartController
- cholesterolChartView: CholesterolChartView
- patientsView: PatientsView
- patientUpdater: PatientUpdater
+ buildViews(): void
- buildGraph: void
- buildSizeOrder(int max): void
- buildCategoryDataset(CategoryDataset dataset): JFreeChart
- createDataSet(CategoryDataset): CategoryDataset
- countValidData(): int
+ update(): void
+ seriesChanged(SeriesChangeEvent e): void
PatientsController
- patientsView: PatientsView
- dbModel: DBModel
- tcChartTimer: Timer
- patientUpdater: PatientUpdater
- createPatients(List<String> patientIds): void
+ update(): void
TCTableController
- patientsView: PatientsView
- tcTableView: TCTableView
- dbModel: DBModel
- patientUpdater: PatientUpdater
- calculateTCAverage(): void
- convertDateToString(Date date): String
- addNewPatientAddedToPatient(Patient patient): boolean
+ update(): void
- addToTCTable(List<Patient> p): void
BPTableController
- patientsView: PatientsView
- bpView: BloodPressureTableView
- dbModel: DBModel
- patientUpdater: PatientUpdater
- addToBPTable(List<Patient> p): void
- convertDateToString(Date date): String
- addNewPatientAddedToPatient(Patient patient): boolean
- addHighSystolicBPObs(List<?extPatient>, ?extPatient): void
- getHighSystolicBPThreshold(): List<Patient> - getHighSystolicBPPatients(double systolicBPThreshold): List<Patient>
- updateHighSystolicBPTracker(): void
- updateHighDiastolicBPTracker(): void
+ update(): void
LoginView
- rootPanel: JPanel
- loginButton: JButton
- practiceInfo: JTextField
- fetchNewEncountersCheckBox: JCheckBox
- fetchNewObservationsCheckBox: JCheckBox
+ getID(Text): String
+ addLoginListenerActionListener listenForLoginButton(): void
+ getPracticeInfo(): boolean
+ fetchNewObservations(): boolean
+ displayErrorMessage
SBPGraphView
- mainPanel: JPanel
- tabPane: JTabbedPane
- patientName: String
- patientPerValues: ArrayList<Double>
+ initChart(String name, ArrayList<Integer> values): void
- buildPanel(): void
- createDataSet(): CategoryDataset
+ showView(): void
CholesterolChartView
- scroller: JScrollBar
- frame: JFrame
- cholesterolPanel: JPanel
- chart: JFreeChart
- scrollPanel: JPanel
- dataset: SlidingCategoryDataset
- patientList: List<Patient>
PatientsView
- patientsMonitorSettings: JTabbedPane
- rootPanel: JPanel
- patientList: JList<Patient>
- addPatient: JButton
- queryTimeTxt: JTextField
- queryBtn: JButton
- setCholesterolBtn: JPanel
- setDiastolicBtnBtn: JButton
- systolicBPTxt: JTextField
- setSystolicBtnBtn: JButton
- monitorCholesterolBtn: JRadioButton
- monitorBPBtn: JRadioButton
- defaultPatientLst: DefaultJtsModelPatient
- hPracid: String
- updateFrshWd: boolean
+ addTabPane(String tabName, JPanel, newPanel): void
+ monitorCholesterol(): boolean
+ monitorBP(): boolean
+ addTabPanelLstPatient patient): void
+ getHPracid(): String
+ setHPracid(String): String
+ addMonitorBtnListenerActionListener listenForMonitorBtn(): void
+ addQueryBtnListenerActionListener listenForQueryBtn: void
+ addSetCholesterolBtnActionListener listenForSystolicBPBtn): void
+ addDiastolicBPBtnListenerActionListener listenForDiastolicBPBtn(): void
+ displayErrorMessage(String errorMessage): void
CholesterolTableView
- tcTable: JTable
- tcTableModel: DefaultTableModel
- monitoredPatients: ArrayList<Patient>
- patientID: JLabel
- patientDOB: JLabel
- patientGender: JLabel
- patientCountry: JLabel
- patientCity: JLabel
- patientStatus: JLabel
- TCTableMonitor: JPanel
- avgCholesterol: double
+ addPatientToMonitor(Patient patient): void
+ removePatientFromMonitor(Patient patient): void
+ updateCholesterolColumn(): void
+ setPatientInfo(String patientDOB, String patientGender, String patientCountry, String patientCity, String patientStatus): void
+ insertPatient(int): void
+ setMonitorBtnListenerActionListener listenForRemoveBtn(): void
+ setTcTableValueAt(Object object, int row, int col): void
+ addTableListenerListSelectionListener listenForTableClick): void
BloodPressureTableView
- bpTable: JTable
- bpTableModel: DefaultTableModel
- monitoredPatients: ArrayList<Patient>
- patientID: JLabel
- highSystolicBP: JPanel
- removeBtn: JButton
- patientDOB: JLabel
- patientSex: JLabel
- patientGender: JLabel
- patientCountry: JLabel
- patientCity: JLabel
- patientStatus: JLabel
- generateBpButton: JButton
- tabPanel: JScrollPane
- bpMonitorPanel: JPanel
- diastolicTracker: JScrollPane
- highSystolicPatients: ArrayList<Patient>
- highDiastolicPatients: ArrayList<Patient>
- systolicBP: double
- diastolicBP: double
+ setPatientInfo(String patientDOB, String patientGender, String patientCountry, String patientCity, String patientStatus): void
+ insertPatient(int): void
+ setBpTableValueAt(Object object, int row, int col): void
+ addToHpDiastolicBPObs(TestPane testPane): void
+ addToHpSystolicBPObs(TestPane testPane): void
+ clearHighSystolicBPObs(): void
+ clearHighDiastolicBPObs(): void
+ setBpTableValueAt(Object object, int row, int col): void
+ addRowToBPTable(Object[] objects): void
+ setHighSystolicBPPatients(Patient patient): void
+ setHighDiastolicBPPatients(Patient patient): void
+ updateSystolicColumn(): void
+ updateDiastolicColumn(): void
+ addRemoveBtnListenerActionListener listenForRemoveBtn(): void
+ setBpTableListenerListSelectionListener listenForTableClick): void
+ addMonitorHighSystolicListenerActionListener listenForHighSystolicTracker(Btn): void
+ addMonitorHighDiastolicListenerActionListener listenForHighDiastolicTracker(Btn): void
+ displayErrorMessage(String errorMessage): void
CholesterolRenderer
- cholesterolAvg: double
+ getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column): Component
SystolicBPRenderer
- systolicBP: double
+ getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column): Component
DiastolicBPRenderer
- diastolicBP: double
+ getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column): Component
Patient
- id: String
- familyName: String
- givenName: String
- dob: Date
- gender: String
- city: String
- state: String
- country: String
- totalCholesterol: double
- latestCholesterolDate: Date
- systolicBPs: List<Object[]>
- diastolicBPs: List<Object[]>
RunApp
+ main(String[] args): void
DBModel
- patientDAO: PatientDAO
- encounterDAO: EncounterDAO
- monitorDAO: MonitorDAO
- observationDAO: ObservationDAO
- practiceInfoDAO: PracticeInfoDAO
+ onStart(String hPracid, boolean fetchEncounters, boolean fetchObs): void
+ getPatientsLst(String hPracid): ArrayList<String>
+ updateCholeObs(String patientId): void
+ getPatientID(Text): String
+ getPatientName(String patientID): String
+ getPatientLastName(String patientID): String
+ getPatientAddressCity(String patientID): String
+ getPatientAddressDate(String patientID): String
+ getPatientAddressState(String patientID): String
+ getPatientAddressCountry(String patientID): String
+ getPatientBirthdate(String patientID): String
+ getPatientGender(String patientID): String
+ getPatientLatestCholesterol(String patientID): double
+ getPatientLatestCholesterolDate(String patientID): Date
+ getPatientSystolicBPs(String patientID, int count): List<Object[]>
+ getPatientDiastolicBPs(String patientID, int count): List<Object[]>
<<interface>>
PatientDAO
+ getPatientName(String patientID): String
+ getPatientFirstName(String patientID): String
+ getPatientLastName(String patientID): String
+ getPatientAddressCity(String patientID): String
+ getPatientBirthdate(String patientID): String
+ getPatientAddressDate(String patientID): String
+ getPatientAddressState(String patientID): String
+ getPatientAddressCountry(String patientID): String
+ getPatientGender(String patientID): String
+ getPatientLatestCholesterol(String patientID): double
+ getPatientLatestCholesterolDate(String patientID): Date
+ getPatientSystolicBPs(String patientID, int count): List<Object[]>
+ getPatientDiastolicBPs(String patientID, int count): List<Object[]>
+ insertPatient(String patientID): void
<<interface>>
PractiseInfoDAO
+ getPracticeIdentifier(String hPracid): String
+ getPracticeNs(String hPracIdentifier): ArrayList<String>
+ insertPracBy(String hPracid): void
<<interface>>
ObservationDAO
+ insertPatientObsByCode(String patientid, String code, String count): void
+ insertCholesterolObs(String patientid, int count): void
+ insertBPObs(String patientid, int count): void
+ getCholesterolObs(String patientid, int count): List<Object[]>
+ getSystolicBPObs(String patientid, int count): List<Object[]>
+ getDiastolicBPObs(String patientid, int count): List<Object[]>
<<interface>>
EncounterDAO
+ getPatientsByPracid(ArrayList<String> hPracids): ArrayList<String>
+ insertEncounter(String encounterid): void
+ insertEncountersByPrac(String identifier, PractiseInfoDAO practiseInfoDAO): void
Mongo
- db: MongoDatabase
- DATABASE_NAME: String
+ connect(): void
JsonReader
+ readAllReader rdr: String
+ readJsonFromUrl(String arg0, JSONObject
PatientRepository
- db: MongoDatabase
- rootUrl: String
+ getAllPatientIds(): ArrayList<String>
+ insertPatient(String patientID): void
+ getPatientFirstName(String patientID): String
+ getPatientLastName(String patientID): String
+ getPatientGender(String patientID): String
+ getPatientBirthdate(String patientID): String
+ getPatientAddressDate(String patientID): String
+ getPatientAddressCity(String patientID): String
+ getPatientAddressState(String patientID): String
+ getPatientAddressCountry(String patientID): String
+ getPatientIdsSortedArray(ArrayList<String> patientIds): String
+ insertPatient(String patientID): void
PractiseRepository
- db: MongoDatabase
- rootUrl: String
+ getPracticeIdentifier(String hPracid): String
+ getPracticeNs(String hPracIdentifier): ArrayList<String>
+ insertPracById(String hPracid): void
ObservationRepository
- db:MongoDatabase
- rootUrl: String
- observationCodes: Map<String, String>
+ insertPatientObsByCode(String patientid, String code, String count): void
+ insertCholesterolObs(String patientid, int count): void
+ insertBPObs(String patientid, int count): void
+ getCholesterolObs(String patientid, int count): List<Object[]>
+ getSystolicBPObs(String patientid, int count): List<Object[]>
+ getDiastolicBPObs(String patientid, int count): List<Object[]>
- insertObs(String obsid): void
- getObsDate(String patientid, String bpType, int count): List<Object[]>
- getLRecentDate(String obsid): Date
EncounterRepository
- db: MongoDatabase
- rootUrl: String
+ getPatientsByPracid(ArrayList<String> hPracids): ArrayList<String>
+ insertEncounter(String encounterid): void
+ insertEncountersByPrac(String identifier, PatientDAO patientDAO, PractiseInfoDAO practiseInfoDAO): void
com.mongodb
Thrift Server
Thrift Server
java swing