Class Diagram The two packages below are from the HAPI FHIR API's core library, the module hapi-fhir-base (https://hapifhir.io/hapi-fhir/apidocs/hapi-fhir-base/). From the description in https://hapifhir.io/hapi-fhir/docs/introduction/modules.html The package below is a module from the HAPI FHIR core library, "This is the core HAPI FHIR library and is always required in order to use the framework. It contains the context, parsers, The class diagram of the application is based on the passive model of the hapi-fhir-structures-r4 and other support classes." (https://hapifhir.io/hapi-fhir/apidocs/hapi-fhir-structures-r4/) It contains the classes (resources) for Encounter, Observation, Practitioner and The two classes FhirContext and IGenericClient allow us to use the methods to fetch and process the requests for Our design might not strictly adhere to the pattern and might have Patient. A collection of any of these resources is known as a Bundle Encounter, Observation, etc with the root URL violated certain design principles. However, we made sure that task2.application (model) is separate from the GUI ca.uhn.fhir.context org.hl7.fhir.r4.model ca.uhn.fhir.rest.client.api.IGenericClient Controller FhirContext **IGenericClient** AddPatientController - practitionerName: String AppController - practitionerIdentifier: String - inputPatientNo: JTextField idInput: JTextField - patientInfoList: List<PatientInfo> - averageTotalCholesterol: double - model: DefaultTableModel + actionPerformed(e: ActionEvent): void - count: int - frame: JFrame + actionPerformed(e: ActionEvent): void + getAverage(): double PatientNameController RemovePatientController - practitionerName: String - practitionerIdentifier: String - table: JTable - patientInfoList: List<PatientInfo> - model: DefaultTableModel - position : int - frame: JFrame + getList(): List<PatientInfo> + getPatientName(): String + actionPerformed(e: ActionEvent): void task2.application ServerBase javax.swing fhirContext: FhirContext java.awt + getClient():IGenericClient EncounterProcessor PatientProcessor - encounterResponse: Bundle patientInfoList: - patientIDs: LinkedHashSet<String> LinkedHashSet<PatientInfo> - searchString: String - patientIDs: LinkedHashSet<String> getResponse(): void + retrievePatientIDs(practitionerName: + processPatients(): String): LinkedHashSet<String> LinkedHashSet<PatientInfo> - returnNextPage(): void ViewData **ObservationSubject** AppLogin <<interface>> contentPane: JPanel Observer table: JTable frame: JFrame - observers: List<Observer> · inputPatientNo: JTextField idInput: JTextField keeps track of - inputFrequency: JTextField attachObserver(observer: Observer): + updateCholesterol(cholesterol: - count: int + main(): void String): void - averageTotalCholesterol: · initialize(): void + detachObserver(observer: Observer): + updateDateTime(dateTime: double String): void + notifyObserver(cholesterol: String, dateTime: String): void + ViewData(practitioner: + update(): void Practitioner): void ButtonColumn PatientInfo table: JTable action: Action - name: String mnemonic: int ViewPatientDetails - surname: String ObservationProcessor originalBorder: Border - id: String focusBorder: Border patientInfoList: List<PatientInfo> patientInfo: PatientInfo birthDate: String The ButtonColumn class is taken from the website renderButton: JButton **⋖** updates index: int - cholesterol: String - gender: String http://www.camick.com/java/source/ButtonColumn.java editButton: JButton - effectiveDateTime: String - city: String editorValue: Object + generateDetails(): void - state: String isButtonColumnEditor: boolean - country: String + processObservation(patient: - cholesterol: String PatientInfo): void getTableCellEditorComponent(- cholesterolDateTime: String table: JTable, value: Object, isSelected: boolean, row: int, column: int): Component getTableCellRendererComponent(table: JTable, value: Object, isSelected: With reference to the UML relations described in boolean, hasFocus: boolean, row: int, column: int): https://www.hl7.org/fhir/practitioner.html https://www.hl7.org/fhir/encounter.html https://www.hl7.org/fhir/observation.html + actionPerformed(e: ActionEvent): void https://www.hl7.org/fhir/patient.html