

Patient Allergy Files

- **Details.cshtml** (Razor HTML page) - This is one of the biggest pages in our application. It includes all code to build the patient interface and plug all of our partial pages into their respective *div* tags.
 - **_tab_Allergy.cshtml** (partial Razor page) - This populates the allergy tab on the details page with HTML and Razor code. This is called using this code: `@Html.Partial("_tab-Allergy", Model)`
 - **Javascript Files at Presentation/Scripts/**
 - **jquery.js** - jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. All of our javascript code depends on this library.
 - **jquery-ui.js** - jQuery UI library, it implements all of our neat animations and widgets
 - **ckeditor.js** - implements a rich text editor for the notes interface
 - **jquery-jvert-tabs-1.1.4.js** - Sets up the tabbed interface
 - **jquery.validate.min.js** - jQuery plugin to support validating forms
 - **jquery.form.js** - jQuery plugin for easier form scripting
 - **jquery.ui.datepicker-en-GB.js** - jQuery datepicker localization file (Ghana uses Great Britain localization files)
 - **jquery.ui.timepicker.js** - jQuery plugin for a timepicker
 - **PrintLabelPatient.js** - support for printing labels
 - **PatientAllergy.cs** - This is the object that implements the many to many relationship from patient to allergy
 - **Allergy.cs** - The class object that holds allergy information. This is just a normal C# class.
 - **IPatientAllergyRepository.cs** - Patient allergy interface empty at this time.
 - **IPatientRepository.cs** - Contains Patient object list interfaces for patient phone, first name, middle name, last name, date of birth and old physical record.
 - **UnitOfWork.cs** - Manages unit of work session (session management). Contains setters and getters for unit of work object.
 - **NHibernate** - NHibernate is an Object-relational mapping (ORM) solution for the Microsoft .NET platform: it provides a framework for mapping an object-oriented domain model to a traditional relational database. Its purpose is to relieve the developer from a significant portion of relational data persistence-related programming tasks.
 - **hibernate.efg.xml** - XML file that contains connection properties to NHibernate. Contains connection string and passwords to database
 - **PatientAllergy.hbm.xml** - Contains xml mappings for PatientAllergy table.

Maps objects to database table rows.

- **Patient.hbm.xml** - XML file that maps patient object to Patient table in the database. Maps all patient personal information info and “bag” collections that contain immunization, allergy and medication etc.
- **PatientController.cs** - Links patientRepository, addressRepository with each other and controls what pages to show to the client depending on the URL. It also was the server code behind our AJAX calls.