**Brief look at how Marcin’s AspergillosisEPR Database works:**

**Uses Identity API for Authorization/Authentication**

HomeController has an authorize attribute, so input is re-routed to AccountController Login Action

Once authenticated program runs HomeController Index action which jumps to Javascript as below:

Jumps into Java script from \Views\Home\Index.cshtml

@Html.Partial("~/Views/Patients/Index.cshtml")

@section Scripts {

<script type="text/javascript">

Patients.bindPatientsModals();

</script>

}

This calls another partial view \views\Patients\Index.cshtml

Marcin makes use of the DataTables.js which is a jquery plugin to format and page data in javascript rather than use MVC to display the data.

This sets up partial view (\_DataTable.cshtml) with table called patients\_datatable which has table headers but empty table body.

Most view are partial view and combine with Views\Shared\\_Layout.cshtml to produce html form.

Marcin also makes extensive use of the Smartadmin template for laying out form formatting.

It also runs some javascript.

Marcin’s javascript files (like Patient.js) are supplied as self invoking functions, which act as namespace for some specialised functions but expose other methods as an interface to the main program.

eg. Patients.js defines an exposed method bindPatientsModals which sets the Jquery event handlers for various events and buttons on the main Patients.html form.

bindPatientsModals in Patients.js (wwwroot/js/Patients.js

bindPatientsModals: function() {

newPatientsModalShow();

bindPatientDetailsShow();

bindPatientEdit();

bindOnDeletePatientClick();

deletePartialFromPopup();

deleteDbPartialFromPopup();

onPatientStatusChange();

onModalClose();

initPatientsDateTimePickers();

bindNewPartialOnPatientFormClick();

onExportOptionsShow();

},

Modal forms are shown in:

<div id="modal-container"></div>

\_Layout.cshtml

@using AspergillosisEPR.Helpers;

<!DOCTYPE html>

<html class="smart-style-6">

@Html.Partial("/Views/Shared/\_Head.cshtml")

<body class="pace-done smart-style-6 hidden-menu" data-action="toggleMenu">

@Html.Partial("/Views/\_ViewImports.cshtml")

<header id="header">

<span class="AntiForge"> @Html.AntiForgeryToken() </span>

@if (User.Identity.IsAuthenticated)

{

await Html.RenderPartialAsync("\_Header");

<section id="top-right">

@(IsSectionDefined("topright") ? RenderSection("topright", false) : Html.Partial("\_TopRight"))

</section>

}

</header>

@if (User.Identity.IsAuthenticated)

{

if (User.IsInRole("Anonymous Role") && !User.IsInRole("Read Role"))

{

await Html.RenderPartialAsync("\_AnonymousAside");

}

else

{

await Html.RenderPartialAsync("\_Aside");

}

}

<div id="main" role="main">

@if (User.Identity.IsAuthenticated)

{

await Html.RenderPartialAsync("\_Ribbon");

}

<div id="content">

<div class="col-xs-12 col-sm-12 col-md-12 col-lg-12">

<h1 class="page-title txt-color-blueDark">

<i class="fa-fw fa fa-home"></i>

<span>

@ViewData["Title"]

</span>

</h1>

@Html.Partial("\_FlashMessage")

@RenderBody()

</div>

</div>

<div id="modal-container"></div>

</div>

@if (User.Identity.IsAuthenticated)

{

<div id="current-user-roles" data-roles="" data-id=""></div>

}

@{ await Html.RenderPartialAsync("\_Scripts"); }

@{ RenderSection("scripts", false); }

<script>

jQuery(function () {

CaseReportForms.showCRFModal();

CaseReportForms.onOptionGroupSelectChange();

})

</script>

</body>

</html>

is-active-route is Angula

fa fa-lg fa-fw are Font Awesome tags

Layout shows a series of PartialViews

\_Header

\_TopRight

\_Aside

\_Ribbon

\_Scripts

\_Scripts runs a jquery script App.init from App.JS Is this the main entry point to start app?

init: function () {

onLogoutLinkClick();

Patients.init();

Settings.init();

Users.init();

Imports.init();

DiagnosisTypes.init();

SideEffects.init();

Charts.init();

Search.init();

Radiology.init();

PatientVisits.init();

CaseReportForms.init();

AnonPatients.init();

MedicalTrials.init();

}

Patients.Init loads patient data ?

init: function() {

initPatientsDataTable();

submitNewPatient();

enableAntiForgeryProtectionWithAjax();

initPatientsDataTable

This uses the DataTables Table plug-in for jQuery.

window.patientsTable = $("#patients\_datatable").DataTable(

This sets up the table patients\_datatable with fields and uses Ajax data. DataTables calls controller DataTablePatientsController action Load to populate the data.

The controller DataTablePatientsController is inherits from DataTable controller, and is a quite convoluted series of calls between the two controllers to return Ajax JSON data with Linq queries.

DataTables controller reads the parameters from the DataTable using the HttpContext data in the call to control how data is returned to Datatable. Sorts paging and record counts etc.

"ajax": {

"url": "/DataTablePatients/Load",

"type": "POST",

"datatype": "json"

Does this use Ajax to load data ?

Effectively run DataTablePatientsController Load

Which seems to load a PatientDataTableViewModel

The table is then rendered from the Javascript code.

Database also uses Rabbitmq message queueing system.