Developer Manual

IU Committee

November 13, 2012

Team Banana

Justin Ashdown

Eric Grounds

Joel Haubold

Jared Short

Dung Truong

Table of Contents

1.0 Introduction 3
1.1 Definitions 3
2.0 Models 3
2.1 Meta data classes 3
3.0 Views 4
3.1 _Layout.cshtml 4
4.0 Controllers 4
4.1 AccountController.cs 4
4.2 AuditLogController.cs 4
4.3 CommDocumentController.cs 4
4.4 CommMemberController.cs 4
4.5 CommitteeSuperAdminsController.cs 4
4.6 CommitteesControler.cs 4
4.7 DiscItemControllers.cs 4
4.8 DiscussionController.cs 4
4.9 DivisionController.cs 5
4.10 MeetingsController.cs 5
4.11 Reports.cs 5
4.12 Search.cs 5
5.0 Database connection 5
6.0 Authentication 5
6.1 Authentication with CAS 5
6.2 Permission routines 5
7.0 Audit log 6
8.0 App_Start 6
8.1 RouteConfig.cs 6

List of Figures

No table of figures entries found.

1.0 Introduction

This project uses the MVC4 Framework.

The models were created using the Entity Framework 5.0.

The controllers and views were created using the builtin controller create function, but were then heavily modified to meet the software specifications.

1.1 Definitions

Division

a general term for a University, Campus, School, or Unit.

2.0 Models

The models listed under Models.tt were automatically generated using Entity Framework 5.0. The models are generated based on the information retreived from the database. To access the database an instance of the jashdownEntities class is created. This class has a collection of objects for each model that is based on a table in the database. Access to the database is done through this class.

This instance of jashdownEntities is usually called db in the source files. To retrieve an tuple from the database like db.Comm.Find(primaryKey1, primaryKey2) is used. This find statement returns a reference to a Comm object retrieved from the database with the passed in primary keys. Collections of tuples can be returned using a similar statement.

CommDocumentPKWithFilenameTags.cs and DiscItemDocumentsWithoutImage.cs classes in this folder are used for listing documents without retrieving the image as well.

GenericDivision.cs contains a model for generalizing Universities, Campuses, Schools, and Units so they can be displayed more easily.

Menu.cs contains the logic to populate the system menu that appears on the left side of the web page.

enumYN.cs is used to validate Yes/No fields.

2.1 Meta data classes

The Meta data classes located in the Models\ModelMetaClasses folder contain metadata for specifying validation constraints. This metadata is used by the model's validate method. The metadata is maintained in a separate file since the Model class file were automatically generated and any code added to these files will be lost if the models need to be regenerated.

MSDN article on metadata "buddy" classes

3.0 Views

3.1 _Layout.cshtml

This file is like the master page or master template. It displays the side menu and web site banner/header. The html generated by the individual views is included in this file by the @RenderBody() function call.

4.0 Controllers

4.1 AccountController.cs

The account controller handles user authentication with the system. It gets the user logged in through CAS, and determines what roles they have in the system. It also handles logging out.

4.2 AuditLogController.cs

The audit log controller is used to add audit logs to the system. It is a static method that any other method can use to add audit logs when needed.

4.3 CommDocumentController.cs

The commDocument controller handles adding, editing, deleting and downloading committee documents. There are views to add, edit and delete.

4.4 CommMemberController.cs

The commMember controller handles adding and editing members to committees. It also handles when a convener selects a committee chair.

4.5 CommitteeSuperAdminsController.cs

The committeeSuperAdmin controller handles adding and editing committee super admins. Only IT admins can do this.

4.6 CommitteesControler.cs

The committees Controler handles the creation, editing, and archival of all committees. Most of the methods are accessible by Committee Super Admins.

4.7 DiscItemControllers.cs

The discItemController houses the methods and logic for creation of the full discussion item. This is mostly accessible only by committee administrators.

4.8 DiscussionController.cs

The discussion controller contains all methods and logic for member interaction within a single discussion. Including commenting and voting.

4.9 DivisionController.cs

This controller displays and gives access to Universities, Campuses, Schools, and Units.

4.10 MeetingsController.cs

The meeting controller contains all methods and logic for Committee Admins to create meeting and maintain meetings.

4.11 Reports.cs

The report controller is used for the creation of reports within the system. Adding additional functionality involved adding a new method and logic, you may need to add additional views as well.

4.12 Search.cs

The search controller is a single point for searching the system. This would be the point for modifying to allow for a wider or deeper search.

5.0 Database connection

The database connection strings are stored in the Web.Config file.

There are two connection strings, both of which point to the same database:

The SecurityConection string is used by the ASP authentication controller.

The jashdownEntities string is used by the jashdownEntities class in Models\Models.Context.cs

6.0 Authentication

6.1 Authentication with CAS

The authentication for the entire system is stored in the Accounts Controller. In order to update how an authentication routine works, edit the functions such as [CommitteeAdmin] or [CommitteeMembe].

In order to use an kind of authentication on a function, simply precede any function declaration with the bracketed “[“ authenticationRoutine “]”.

Currently available methods are…

[CommitteeMember], [CommitteeMemberVoting], [CommitteeAdmin], [CommitteeConvener], [CommitteeSuperAdmin], [ITAdmin], [MeetingMember], [MeetingAdmin], [MeetingConvener], and [MeetingVotingMember].

6.2 Permission routines

The routines in AccountController.cs file control access. The routines read the url and extract the primary keys. They then retrieve the corresponding tuple in the database and check the currently logged in user against the database information to see if the user has access based on their roles with respected to the requested data.

7.0 Audit log

8.0 App_Start

8.1 RouteConfig.cs

This file contains routing information. There are six routes that are used to route the url request to the correct controller and to pass data to the controller. The urls follow the following pattern, controller/action/primaryKey1/primaryKey2/…/primaryKey6. The primary keys are optional depending on the controller and action. This is best explained with an example. Consider the url /committees/details/2/1. ASP MVC will call the details action of the committees controller and pass 2 and 1 as primary key 1 and primary key 2 respectively. The controller will then use these two parameters to lookup the correct committee information. To access a meeting, three primary keys values are needed since a meeting has three parts to the primary key. To create an object the primary key of the parent object is required. Meetings/create/2/1 will try to create a meeting for the committee with the primary key 2,1.