**ENTERPRISE FRAMEWORKS**

**PLANNING APPLICATION MANAGEMENT WEBSITE**

**BLUE TEAM**

**Des Harold**

**Paul Kelly**

**James Larmon**

**Colm Smith**

**OVERVIEW:**

Enterprise Framework project to develop a Planning Application Management System.

Development environment used is the .NET framework

MicroSoft integrated development environment Visual Studio

Architectural pattern used is MVC

API is REST

**APPLICATION:**

**Planning Application Website**

The purpose of the application is to manage planning applications submited by the user. The application allows the user to enter their details such as name and address and submit any plans and drawings they have pertaining to the specific property in question. The user information is stored in a database and drawings are stored in a storage account created in Blob storage on the windows azure cloud. There is also an email azure link to the surveyor of the building which is stored in the blueteam blob storage account.

The information stored can be edited, deleted, displayed and searched.

**DATABASE:**

The application is using the SQL Express package for visual studio as the database to store the user information.

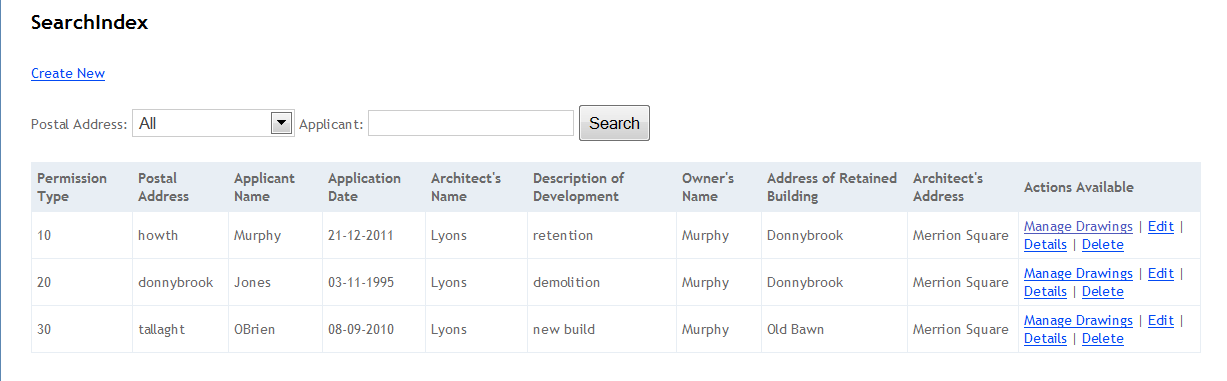
The data can be created, deleted, edited and retrieved.

**/Permission/SearchIndex**

A dropdown box allows the database to be searched by the postal address.

Records can be Edited and Deleted.

Manage Drawings button to show files held in Blob Storage.

****

**SEARCHING THE DATABASE:**

The database can be searched by the applicants postal address for which there is a dropdown box, or by applicant name.

The PermissionController sets the search parameters for the SearchIndex View.

**PermissionController.cs code to set search parameters:-**

public ActionResult SearchIndex( string planAddr, string searchString)

{

var AddrLst = new List<string>();

var AddrQry = from d in db.Permissions

orderby d.postalAddr

select d.postalAddr;

AddrLst.AddRange(AddrQry.Distinct());

ViewBag.planAddr = new SelectList(AddrLst);

var permissions = from m in db.Permissions

select m;

if (!String.IsNullOrEmpty(searchString))

{

permissions = permissions.Where(s => s.appName.Contains(searchString)).Take(3);

}

**Code for dropdown box in the SearchIndex.cshtm View:-**

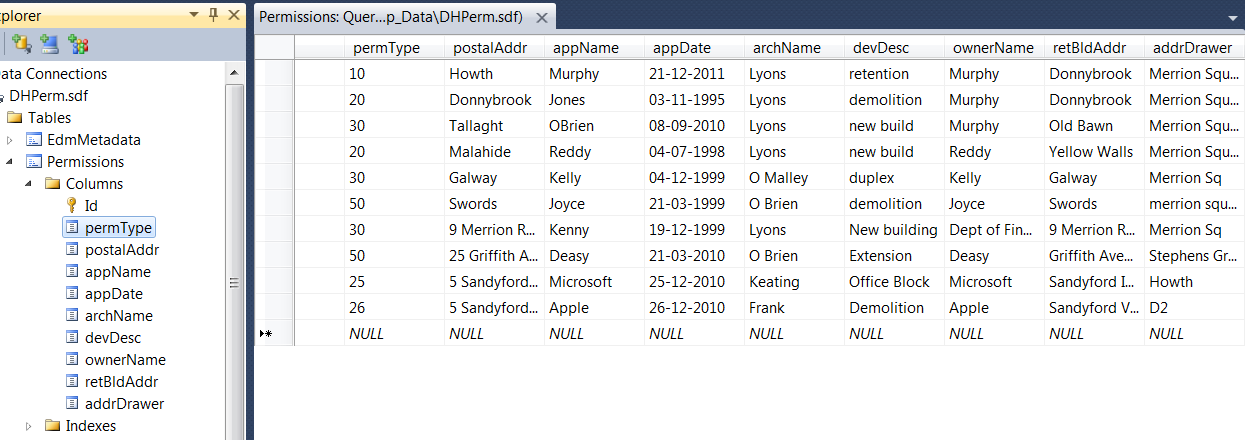
@using (Html.BeginForm()){

<p> Postal Address: @Html.DropDownList("planAddr", "All")

Applicant: @Html.TextBox("SearchString")

<input type="submit" value="Search" /></p>

}



Database view in visual studio showing user records

The database is stored in the App\_Data folder and is called DHPerm.sdf

**STORAGE:**

**Blob Storage:**

In order to store images of maps and plans associated with the project a Blob Storage account was created in windows azure. The storage account for this application is called blueteam. One can use Blob storage to expose data publicly to the world or privately for internal application storage.

The images can be uploaded and retrieved from storage using HTTP or HTTPS. For the purpose of this project the data has been set up as public access.

Individual blobs are stored inside a container.

The container is created inside the DHBlobController.cs

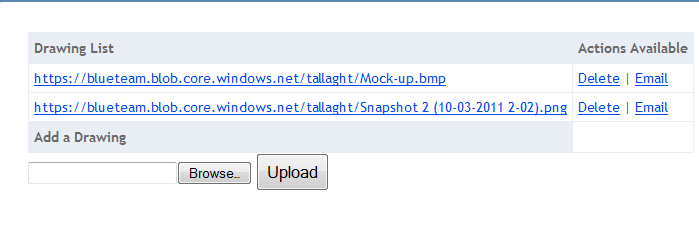
**DHBlob/Index/tallaght**

**Storage account**  - blueteam

**Container** - tallaght

**Blobs** - Mock-up.bmp

- Snapshot 2(10-03-2011 2-02).png

****

**WINDOWS AZURE BLOB STORAGE CONCEPT FOR ABOVE URL:**

ACCOUNT

blueteam

BLOBS

Mockup.bmp

Snapshot2

CONTAINER

tallaght

**Uploading drawing to Blob storage:**

For a drawing to be uploaded to the azure blob storage account it must firstly be contained in a local folder which is refrenced in the **DHBlobController.cs**

// This action handles the form POST and the upload

[HttpPost]

public ActionResult GetFname(HttpPostedFileBase file, string conatinerId)

{

// Verify that the user selected a file

if (file != null && file.ContentLength > 0)

{

// string filePath = Path.GetFullPath(file.FileName);

// extract only the fielname

string fileName = Path.GetFileName(file.FileName);

// store the file inside ~/App\_Data/uploads folder

//var path = Path.Combine(Server.MapPath("~/App\_Data/uploads"), fileName);

string filePath = Path.Combine(@"C:\Drawings", file.FileName);

//save the file to our local path

The drawing is uploaded to a container in the azure storage account which is private by default. To make the drawings accessible to everyone for the purpose of this project the container permissions are set to public.

container.SetPermissions(

new BlobContainerPermissions { PublicAccess = BlobContainerPublicAccessType.Blob });

**MVC STRUCTURE FOR PLANNING PROJECT:**

