# Bitbucket keys password:

FedorLutron2021

# Application overview:

## ViewModel structure:

ViewModel named as QuantumResi in the project contains code for all the applications. It contains modules for all the features in the app. They are organized in a way that follows the structure of the Designer program. (e.g. QuantumResi > Design > Screens or QuantumResi > Programming > Screens).

ViewModels have View specific properties (e.g. what area selected) and every operation has command registered in the corresponding ViewModel. For every action, the command bound in xaml is used instead of the event that is handled in the xaml. After that, the delegate is used for executing the command.

## Command execution procedure:

Command specified in the xaml > Command will then be fired in the cs file > The delegate is fired > The method passed into the delegate is executed > Fires a corresponding ModelView that creates or modifies objects as needed > ModelView communicates with the DomainObject layer

## Structure of the database:

Databases > Project > Tables > DomainObjects Tables.

We have a table for each DomainObject. The properties of the object are columns of the database. Different instances are added as rows in the table. During runtime, it fetches the database and creates DomainObjects on the basis of the data in the database. The communication with the database is done in the region Zone Info through WriterInfoItem.

Databases > Project > Programmability > Stored Procedures > Procedure Tables

Contains stored procedures associated with and can be performed on each object type.

DomainObjects have a lot of design patterns followed. Command Pattern. Visitor Pattern for writing on the database and processor. Data visitor for visiting database and processors. Factories for creating DomainObjects. ModelView is the wrapper for the DomainObjects layer.

# Project and Info Databases Overview:

Lutron.Gulliver.Core.Gonfiguration.dll.config – this file stores all the references to the databases.

## Project DB:

Hierarchy is done through creating the parent-child relation, with relation being a property of the child object i.e. executed through assigning parent ID to the child object. Child and parent objects can be of different types.

WriterInfo class contains information per each object type that will be used by any writer. In calling the class, we specify all the procedures and then all the properties. WriterInfoItem class defines a structure in which the information about each object type is stored in the WriterInfo. InitializeFromDatabase that reads data from sql and loads it into code has to be overwritten from each DomainObject. ‘

Implementation of WriterInfo requires to override InitializeFromDatabase.

To load the data from the database, procedures are normally used. The sql database files are stored in the ProjectDatabase > Create SQLServer Scripts. To add a column, one has to modify the corresponding script (i.e. add entry for CREATE TABLE and update PROCEDURES). DatabaseResources then runs the CreateScripts.xml that contains links to the sql files. To create schema, we have to build the DatabaseResources project into CreateSchemas.sql that is a compilation of all the sql files from the ProjectDatabase.

## ModelInfo:

Keeps data about all the hardware devices. It is static database.

## ReferenceInfo:

Stores data on which products available in which versions of the app and regions. Product settings in different regions.

## Preferences:

Stores user preferences in the GUI.

# Area Type Prediction

TODO: describe the flow of the program and specific classes where it will happen.

Track on focus change event from view > viewModel > model

Create Class Scheme

Relevant files:

<!--

Area Selected ViewMode HierarchicalDataTemplate.

It is using AreaCommonModeTemplate.

and then overriding the visibility keys.

In this mode textblock and edit hyperlink is visible

-->

C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\AreaTreeView.xaml

/// <summary>

/// Select the Area and change mode to Edit

/// </summary>

/// <param name="parameter"></param>

/// <summary>

/// Complete the Edit/Add Mode of the AreaModelView.

/// If Area is in Add mode and Enter key is pressed, create another Area

/// or else just change the mode to View, select the Area and show the AddNewArea button.

/// </summary> C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\AreaTreeViewModel.cs

/// <summary>

/// Edit Command

/// </summary>

C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\TreeViewModel.cs

/// <summary>

/// Gets a unique name of the domain object.

/// </summary>

/// <typeparam name="T"></typeparam>

/// <param name="domainObj">The domain obj.</param>

/// <param name="pasteDomainObjectElements">The paste domain object elements.</param>

/// <param name="cutDomainObjectElements">The cut domain object elements.</param>

/// <param name="domainObjectName">Name of the domain object.</param>

/// <param name="similarObjectCount">Same named objects count.</param>

/// <returns></returns>

protected static string GetUniqueName<T>

C:\code\src\Lutron\Gulliver\ModelViews\ModelViews\AreaModelView.cs

## Edit Area Name Sequence

(AreaTreeView) Binding to EditCommand > (TreeViewModel) EditCommand > (AreaTreeViewModel) RenameArea > the mode of the object AreaModelView (which represents the instance of area) is set to Edit

The edit text box is somehow bound to the Name Field of the DomainObjectModelBase and is changed on every key stroke, which also calls the function called onNameChanged in the Area DomainObject.

(AreaTreeView) Binding to UpdateCommand > (AreaTreeViewModel) UpdateCommand > (AreaTreeViewModel) UpdateArea > the mode of the object AreaModelView is set to ModelViewModes.View > in the same function call the onNameUpdated (a new function) in the AreaModelView > which calls the onNameUpdated (a new function) in the Area DomainObject > that function accesses the Name property of the parent class and sends a request using the APITools > then it updates the AreaTypePredicted (a new field) field.

# API Tool Notes

AwsAuthenticator class extends the IAuthenticator, which is in a has a relation with RestRequest class. ApiClientHelper constructor attaches the AwsAuthenticator to the RestRequest. Then, ApiClientHelper method ExecuteRestRequestAsync uses RestRequest class to asynchronously execute the request.

Parameters: method, headers, parameters, resource, token are all passed into the ApiClientHelper.ExecuteRestRequestAsync method.

Token – from my understanding is just an instance of the CancellationToken class.

Resource – not sure what that is

Headers – used to attach the "x-api-key" header

Parameters – Not quite sure what it is for

Method – it will always be POST method

## Questions

* How to use the API tool.
  + AuraRollingCalController.cs line 1840
* It seems some of sha256 keys were hardcoded. What should and should not be modified?
  + Talk to the ds team about that
  + Used to request the authentication keys in the awshelper
    - Store the returned keys
* What is the process of Authentication? Just in case if something goes wrong, I had an idea where to look at.
* What is the format of the request?
  + Seems like all the headers should be the same.
* Sequence of the request execution? I thought that the token should be request at first, but it seems that I was wrong.
  + Token is not really used
* Who should we talk to get all the keys and modifications for our project?
  + Talks to the DS team
* At what point do you attach the content to the request?
  + ChromalityCheck function GetBody

## Changes to documents:

* Follow example: <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/5499/diff>
* <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/6671/overview>
* C:\code\src\Lutron\Gulliver\ProjectDatabase\Create SQLServer Scripts\Area.sql
  + Added columns for AreaTypePredicted and AreaTypeSelected to the Create table and procedures in 19 places.
* Compiled DatabaseResources project
  + This file was changed
  + C:\code\src\Lutron\Gulliver\Infrastructure\DatabaseFramework\SQLServer\ProjectDatabaseScripts\CreateSchemas.sql
* **TODO**: change CommonAssemblyInfo.cs, but there are a couple
* **TODO**: update DBVersionToGUIVersionMap.xml and DatabaseVersionManager.cs
* How to compile the DB
  + After the compilation of the DatabaseResources project, the GUI should automatically create a new project DB after the new project is created in GUI
* Update Project DB
  + Go to Infrastructure/DatabaseFramework/SQLServer
  + Add conversion scripts for the new database version into the Conversion Folder
  + Add new scripts to Synchronization.sql
    - Change all GO commands with Delimiters
    - Drop procedures before every procedure
    - Remove table from Synchronization.sql?
    - Add columns to existing table. Run script if column doesn’t exist add it.
  + Update Conversion.csproj
  + Update DBVersionToGUIVersionMap.xml
* Added submodule to DomainObjects/CondorApiIntegration
* C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\AreaTreeViewModel.cs
  + Added call to the UpdateArea() on line 12260 call from the UpdateArea() on line 1157.
* Area.cs
  + UpdateArea() on line 13174
  + Added stuff for AreaTypePredicted and AreaTypeSelected properties on lines:
    - 930 – Declared Properties. **They are declared as ints**
    - 21550 – Getters and setters for the properties
    - **TODO** 19276 – ReadProperties() Retrieved as int32
    - 7320 – what is RaisePropertyChangeAndMarkDirty
    - 16044 – Writer class
    - 5632 – InitializeFromDatabase Retrieved as int32
  + Implement PredictAreaTypeCommand
* C:\code\src\Lutron\Gulliver\DomainObjects\WriterInfo.cs
  + Line 313
  + **TODO**: maybe other places, but I don’t think so
* In AreaTreeViewModel, added a call to the UpdateArea function to the UpdateArea function in the AreaModelView.
* In AreaModelView, Added UpdateArea function after the Add Area region
  + Added call to the UpdateArea() on line 12260
* Added UpdateArea method to the Area domain object in the Area Commands region
* Update the ModelInfo DB
  + Example: <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/6307/diff#src/Lutron/Gulliver/InfoObjects/ModelInfo/DimCurveDefinitionInfo.cs>
  + <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/6652/overview>
  + In ModelInfoSQLDatabase:
    - ModelInfoSQLDatabase > dbo > Tables
      * Add new table (LstAreaTypes.sql)
    - ModelInfoSQLDatabase > dbo > StoredProdedures.
      * Add procedure SEL\_AreaTypeInfo and SEL\_ALLAreaTypeInfos.sql
    - ModelInfoSQLDatabase > DataScripts.
      * Create procedure to fill LstAreaTypes.sql
    - ModelInfoSQLDatabase > Deployment scripts
      * Add procedure for creating an enum to ExecuteDataScripts.sql
    - Include to build in ModelInfoSQLDatabase.sqlproj
  + In InfoObjects:
    - InfoObjects > ModelInfo
      * Add class for the table
      * Update ModelInfoControler.cs, add List of the created class and return a ReadOnlyCollection
      * Update ModelInfoObjectType.cs, add enum for created class
    - Update InfoObjects > InfoObjects.csproj
    - InfoObjectsTests > ModelInfoTests
      * AreaTypeInfoTest.cs
* Create new class AreaTypeDetails which is not a domain object. Area class will hold a reference for that class. You can refer ProjectQuotationDetails. Added table tblAreaTypeDetails, without updating the processor database at the moment.
  + Ex: <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/6671/overview>
  + Add instantiation in the constructor for a new area
  + Add FetchOrCreate
  + Add a function PredictAreaType that uses CondorApi, which runs in Task.Run
* Add feature flag:
  + Add Feature flag: <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/6721/overview>
  + Should my feature be feature flagged. How will it work when it is offline. <https://wiki.intra.lutron.com/display/POL/Guidelines+to+apply+Feature+Flagging+in+Desktop+App>
* Added FuzzySharp package to DomainObject
* Added a .json dictionary for fuzzy match algorithm: <https://git.intra.lutron.com/projects/GG/repos/code/pull-requests/5218/overview>
* Add an interface for the PredictAreaType and using Moq plugin implement it for testing

## Added Nuget packages to the project to make Api tool work

* if you are adding an existing package, you just need to right click on the solution name, hit manage nuget packages, and then install it from the screen that comes up
  + Installed into the CondorApiIntegration project
  + RestSharp
  + Newtonsoft.Json
  + AWSSDK.Core.3.7.0.35
  + AWSSDK.SecretsManager.3.7.0.33
  + Microsoft.Extensions.DependencyInjection.Abstractions.2.2.0
  + System.Buffers.4.4.0
  + System.ComponentModel.Annotations.4.5.0
  + System.Numerics.Vectors.4.4.0
  + System.Runtime.CompilerServices.Unsafe.4.5.1
  + System.Memory.4.5.1
  + Microsoft.Extensions.Primitives.2.2.0
  + Microsoft.Extensions.Caching.Abstractions.2.2.0
  + Microsoft.Extensions.Options.2.2.0
  + Microsoft.Extensions.Caching.Memory.2.2.0
  + AWSSDK.SecretsManager.Caching.1.0.3

## Condon API Integration Discussion

* Read about aws secrets manager
* Keys
  + x-api-key:
  + access:
  + secret key:
  + This is region: "us-east-1"
  + secretID is also hardcoded
* Link format
  + <https://models.dev.data.lutron.io/v1/run/roomtype-prediction-residential/1.0.0>
  + Version might change, update for each version of designer tool
* Request format
  + {"data": ["dylans room"]}
  + GetBody is json
  + GetDataBody json list of area names. Should probably just add a name instead of list of lists
* Response format
  + {"result": [["Bedroom", "Living Room", "Kitchen", "type1", "type2"]]}
  + Right now it is a list of lists
* In authorization:
  + Access key:
  + Secret key:
  + AWS region: us-east-1
  + Service Name: execute-api:
* Headers:
  + X-api-key:

I expect:

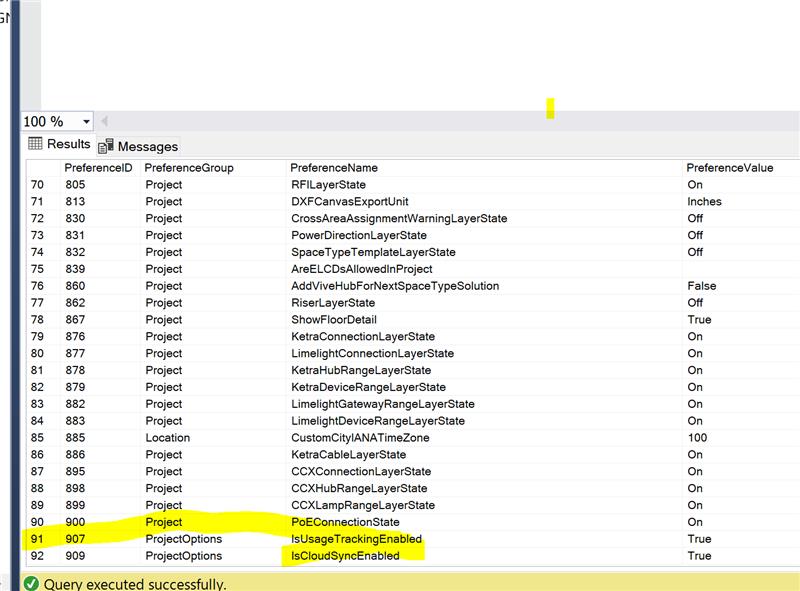
* Development keys
  + x-api-key:
  + access:
  + secret key:
* Request body example
* Response body example

## AreaType vs Caseta AreaCategory

* Talk to Dicsha about how caseta works.
* Merge the list of area type with caseta integer key
* Type of AreaCategory. Is it an integer? Is it nullable? The type should be compatible with the AreaType
* Where the table should live? Try to share with the could. Lives in one place

* <https://git.intra.lutron.com/projects/LP/repos/sqlite-database/browse/lutron-db-skeleton-dump.sql#4173-4257>
* Foreign key constraint. Rockcopper database
* Use it as a config file resource?
* Nothing with processor, just GUI.

## Opt-in data discussion

* Project db
* Firoze Khan is in charge of that
* it is stored in [tblPreferences]
* 

## New login for API

This new login endpoint should be available in our dev environment by the end of the week or so - I will provide a URL - the short story is make a POST request to the URL with the user's login token set in the '*x-auth-token'* header - the endpoint will respond with JSON (as a string) that looks like:

{

"aws\_api\_key": "somevalue",

"aws\_access\_key\_id": "somevalue",

"aws\_secret\_access\_key": "somevalue"

}

Those values should then be packaged into the header sent to the model invocation calls.

UM token - User Management - The GUI should have it by virtue of the user logging in to the GUI

<https://models.dev.data.lutron.io/v1/login>

* Getting UM token:
  + string token = UserManager.Instance.SecurityToken;
  + Should be a Production version of ServicesConfig.json file in bin folder
  + used in
  + C:\code\src\Lutron\Gulliver\Infrastructure\Services\myProjects\myProjectsServiceManager.cs
* Check if anonymous data is enabled
  + (bool?)PreferenceManager.Instance.GetPreferenceValue(PreferenceInfoType.IsUsageTrackingEnabled, PreferenceTypes.Project);

# Lock Issue:

System.AggregateException

HResult=0x80131500

Message=One or more errors occurred.

Source=mscorlib

StackTrace:

at System.Threading.Tasks.Task.ThrowIfExceptional(Boolean includeTaskCanceledExceptions)

at System.Threading.Tasks.Task.Wait(Int32 millisecondsTimeout, CancellationToken cancellationToken)

at System.Threading.Tasks.Task.Wait()

at Lutron.Gulliver.DomainObjects.Area.UpdateArea() in C:\code\src\Lutron\Gulliver\DomainObjects\Area.cs:line 13184

at Lutron.Gulliver.ModelViews.AreaModelView.UpdateArea() in C:\code\src\Lutron\Gulliver\ModelViews\ModelViews\AreaModelView.cs:line 12266

at Lutron.Gulliver.QuantumResi.Common.AreaTreeViewModel.UpdateArea(Object parameter) in C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\AreaTreeViewModel.cs:line 1169

at Lutron.Gulliver.UIFramework.WpfCore.DelegateCommand`1.Execute(T parameter) in C:\code\src\Lutron\Gulliver\UIFramework\WpfCore\Behaviours\DelegateCommand.cs:line 207

at Lutron.Gulliver.UIFramework.WpfCore.DelegateCommand`1.System.Windows.Input.ICommand.Execute(Object parameter) in C:\code\src\Lutron\Gulliver\UIFramework\WpfCore\Behaviours\DelegateCommand.cs:line 331

at Lutron.Gulliver.QuantumResi.Common.AreaTreeView.EditableControl\_IsKeyboardFocusWithinChanged(Object sender, DependencyPropertyChangedEventArgs e) in C:\code\src\Lutron\Gulliver\QuantumResi\Common\AreaTree\AreaTreeView.xaml.cs:line 135

at System.Windows.UIElement.RaiseDependencyPropertyChanged(EventPrivateKey key, DependencyPropertyChangedEventArgs args)

at System.Windows.FocusWithinProperty.FireNotifications(UIElement uie, ContentElement ce, UIElement3D uie3D, Boolean oldValue)

at System.Windows.ReverseInheritProperty.FirePropertyChangeInAncestry(DependencyObject element, Boolean oldValue, DeferredElementTreeState treeState, Action`2 originChangedAction)

at System.Windows.ReverseInheritProperty.FirePropertyChangeInAncestry(DependencyObject element, Boolean oldValue, DeferredElementTreeState treeState, Action`2 originChangedAction)

at System.Windows.ReverseInheritProperty.OnOriginValueChanged(DependencyObject oldOrigin, DependencyObject newOrigin, IList`1 otherOrigins, DeferredElementTreeState& oldTreeState, Action`2 originChangedAction)

at System.Windows.Input.KeyboardDevice.ChangeFocus(DependencyObject focus, Int32 timestamp)

at System.Windows.Input.KeyboardDevice.PostProcessInput(Object sender, ProcessInputEventArgs e)

at System.Windows.Input.InputManager.RaiseProcessInputEventHandlers(ProcessInputEventHandler postProcessInput, ProcessInputEventArgs processInputEventArgs)

at System.Windows.Input.InputManager.ProcessStagingArea()

at System.Windows.Input.InputManager.ProcessInput(InputEventArgs input)

at System.Windows.Input.InputProviderSite.ReportInput(InputReport inputReport)

at System.Windows.Interop.HwndKeyboardInputProvider.ReportInput(IntPtr hwnd, InputMode mode, Int32 timestamp, RawKeyboardActions actions, Int32 scanCode, Boolean isExtendedKey, Boolean isSystemKey, Int32 virtualKey)

at System.Windows.Interop.HwndKeyboardInputProvider.PossiblyDeactivate(IntPtr hwndFocus)

at System.Windows.Interop.HwndKeyboardInputProvider.FilterMessage(IntPtr hwnd, WindowMessage message, IntPtr wParam, IntPtr lParam, Boolean& handled)

at System.Windows.Interop.HwndSource.InputFilterMessage(IntPtr hwnd, Int32 msg, IntPtr wParam, IntPtr lParam, Boolean& handled)

at MS.Win32.HwndWrapper.WndProc(IntPtr hwnd, Int32 msg, IntPtr wParam, IntPtr lParam, Boolean& handled)

at MS.Win32.HwndSubclass.DispatcherCallbackOperation(Object o)

at System.Windows.Threading.ExceptionWrapper.InternalRealCall(Delegate callback, Object args, Int32 numArgs)

at System.Windows.Threading.ExceptionWrapper.TryCatchWhen(Object source, Delegate callback, Object args, Int32 numArgs, Delegate catchHandler)

This exception was originally thrown at this call stack:

[External Code]

Inner Exception 1:

AggregateException: One or more errors occurred.

Inner Exception 2:

LockRecursionException: A read lock may not be acquired with the write lock held in this mode.

## Other shit

System.InvalidOperationException

HResult=0x80131509

Message=The calling thread cannot access this object because a different thread owns it.

Source=WindowsBase

StackTrace:

at System.Windows.Threading.Dispatcher.VerifyAccess()

at System.Windows.Application.get\_MainWindow()

at Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowDialogBox(String message, String caption, CustomDialogButtons customDialogButtons, CustomDialogIcons icon, CustomDialogProminentButton prominentButton, Boolean hideCloseButton) in C:\code\src\Lutron\Gulliver\UIFramework\WpfCore\Services\DialogServices\Implementation\MessageDialogService.cs:line 670

at Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowDialogBox(String message, String caption, CustomDialogButtons customDialogButtons, CustomDialogIcons icon) in C:\code\src\Lutron\Gulliver\UIFramework\WpfCore\Services\DialogServices\Implementation\MessageDialogService.cs:line 650

at Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowError(String message) in C:\code\src\Lutron\Gulliver\UIFramework\WpfCore\Services\DialogServices\Implementation\MessageDialogService.cs:line 82

at Lutron.Gulliver.QuantumResi.Main.ShellViewModel.CreateNewProject(Boolean isCreatedFromMyProjects) in C:\code\src\Lutron\Gulliver\QuantumResi\Main\Shell\ShellViewModel.cs:line 2375

at Lutron.Gulliver.QuantumResi.Main.ShellViewModel.<>c\_\_DisplayClass271\_0.<CreateNewProjectAndDisableAutoSave>b\_\_0() in C:\code\src\Lutron\Gulliver\QuantumResi\Main\Shell\ShellViewModel.cs:line 2288

at System.Threading.ThreadHelper.ThreadStart\_Context(Object state)

at System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)

at System.Threading.ExecutionContext.Run(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)

at System.Threading.ExecutionContext.Run(ExecutionContext executionContext, ContextCallback callback, Object state)

at System.Threading.ThreadHelper.ThreadStart()

This exception was originally thrown at this call stack:

[External Code]

Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowDialogBox(string, string, Lutron.Gulliver.UIFramework.WpfCore.Services.CustomDialogButtons, Lutron.Gulliver.UIFramework.WpfCore.Services.CustomDialogIcons, Lutron.Gulliver.UIFramework.WpfCore.Services.CustomDialogProminentButton, bool) in MessageDialogService.cs

Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowDialogBox(string, string, Lutron.Gulliver.UIFramework.WpfCore.Services.CustomDialogButtons, Lutron.Gulliver.UIFramework.WpfCore.Services.CustomDialogIcons) in MessageDialogService.cs

Lutron.Gulliver.UIFramework.WpfCore.Services.MessageDialogService.ShowError(string) in MessageDialogService.cs

Lutron.Gulliver.QuantumResi.Main.ShellViewModel.CreateNewProject(bool) in ShellViewModel.cs

Lutron.Gulliver.QuantumResi.Main.ShellViewModel.CreateNewProjectAndDisableAutoSave.AnonymousMethod\_\_0() in ShellViewModel.cs

[External Code]

# PCAL – Upload data to the database

## Install MySQL workbench:

* <https://wiki.intra.lutron.com/pages/viewpage.action?spaceKey=KET&title=Installing+and+Setting+Up+MySQL+Workbench>

## Add Database Connection

Production Class Db project

DutyCycleTest::TestId --{ DutyCycleTestData::Test\_ID

Mysql data Framework

Inside of the project, there are classes with properties that correspond to table rows. Instance of the class is the row of data in the table

IProductionDatabaseService – add instance of the method for the created classes. Implement in ProductionDatabaseService.

See DutyCycleTest as an example for the process

ProductionCalDbContext

D3CookieDtabaseManager.cs – create database using factory. Call the method created in DutyCycleTest

PR with example of how to set up entities etc.: <https://git.intra.lutron.com/projects/KET/repos/ketramanufacturingtools/pull-requests/545/overview>

### Files that add the entities to the DB context:

* ProductionCalDatabaseTools\Libraries > ProductionCalDb > Entities
  + DutyCycleTest.cs
  + DutyCycleTestData.cs
  + ProductionCalDbContext.cs

### Files that add the method for creating the DB entries:

* ProductionCalDatabaseTools\Libraries > ProductionCalDb > Service
  + IProductionDatabaseService.cs
  + ProductionDatabaseService.cs
    - Note: In the method I added, we pass in a guid as well as the test data object list. No need to do that in your case; we just do that so we can link from a different table to the DutyCycleTest table.

### Example of how to call the method that creates the DB entries:

* Lamp\_Production > CookieTester\CookieTester > CookieEngine
  + D3CookieDatabaseManager.cs
    - Note: Relevant methods here are the constructor, where we instantiate the database service, and WriteToDutyCycleTable()

# Group Presentation

## Bar:

* Bar is not only a functional space. Integral part of design and experience. Art installation
* It is equipped with designer bottles and installations which serve as decorations
* We want to highlight each of those areas and installations with their own light sources
* In our standard package, lights are very much equipped to do that, but they don’t give the full experience.
* Talk about kitkats
* The premium package is equipped with the most suitable tools for the job.

## Restrooms:

* We explore a very similar concept with the bathroom arrangement.
* The standard package will give the lights and all the color and vibrancy that come with them,
* but premium package will have additional highlights in exactly the spots that we want to give the full experience.
  + Kitkats

## Deck:

* Work with natural light as much as possible use the most appropriate tools like d3 to create the natural light experience
* In the pool deck area premium package we are not just trying to light the area and create the mood
* At night also accentuate the most important areas with kitkats to create a relaxing ambiance
* Premium package is all about creating an uncompromised experience

## Supporting Devices:

* Of course all of the lighting requires the brains that would control it.
* We do not compromise anything when it comes to the backend devices.
* The only premium feature that one could get is premium finish for the keypads that control the lighting

## Final Words:

* Ketra lights is all about enhancing and creating experiences that you did not even think were possible.
* It is about immersion and building ambience that sets the mood

# Final Presentation:

## Intro

* agenda/outline

## Problem

* First ML model to be put up for the company
* First one to start working on this proposal
* Also the new potential use case for the area type data is to infer zone layering based on area type + fixture type, so you could mention that as a possible future application

## Condor API

## Good and Bad

* Saurab suggested changes to the way enum and area type details are stored
* Project DB update process

## Manufacturing Tool

* My initiative