Smart Design Plugin

1. **Description:**
    a. Part 1:
        i. Part 1 focuses on taking the user selections and gets it's (ItemNumber, ItemName, Location, Quantity, Vendor, Manufacturer, Model#, Description) parameters and get them in a dummy schedule named "**Smart Schedule**"
    b. Part 2:
        i. Part 2 is about accessing the FileMaker database then:
            1. Extract the data from the specified project to the smart schedule in the Revit model
            2. Editing items extracted then update the data in the FileMaker data base
            3. Add items created in Part 1 to the File maker database

2. **Program outline:**
    a. The program consist of two projects which represented in two buttons named (Create Schedule & Synchronize) :
        i. First the project which represent the button **Create Schedule** which consist of :
            1. **Class command** :
                a. First gets the revit document then get the users selections and extract data from the selected elements (Family ,Type , Room)
                b. Then call the WPF named **"AddItemstoSmartSchedule"** and send the elements groups and informations to it
            2. **WPF "AddItemstoSmartSchedule"** :
                a. The main class call a class named **ClustersData**
                b. Event named **Button_Click** to close the window when needed
                c. Event named **Check_All** to check all items in the Data grid
                d. Event named **Check_None** to uncheck all items
                e. Event named **Check_Selected** to check only selected items
                f. Event named **CreateSchedule** which will :
                    i. Get all the data from the data grid and filter only the checked items and put them in a list

   ii. Call a class named "**CreateSmartSchedule**" to create the schedule which will get if there is already a smart schedule created or not
   iii. If it exists then the data will be got from the schedule to check if the items already exist or not
   iv. If the smart schedule already exists then a WPF named "**SmartScheduleExistError**" will be called

3. **Class Cluster Data:**
   a. This class uses the object oriented programming to create items containing the data of the chooses items (RecordID ,RevitID ,ProjectNumber ,ItemNumber ,Family&Type ,Location ,Quantity ,Vendor ,Manfacturer ,Model ,Description ,Website) then retrieve it to the **WPF "AddItemstoSmartSchedule"** to represent it in the data grid.
4. **Class CreateSmartSchedule :**
   a. This class check if the schedule exist or not if it exists it retrieve this result to the **WPF "AddItemstoSmartSchedule"** if not it will create the smart schedule then fill it with the checked items data
5. **WPF "SmartScheduleExistError" :**
   a. This WPF warns the user that there is already a smart schedule and then transmit to the next WPF named "**Existing_DataSchedule**".
6. WPF "**Existing_DataSchedule**":
   a. The main Event represent the data that already exists in a data grid and give the user the choice wether to replace the exiting items in the smart schedule or all the checked items to the existing items or merge the the checked items to the existing items
   b. The other events send the action and the data of the checked items to the class named "**ActionWithExistingSchedule**"
7. "**ActionWithExistingSchedule**":

        a. This class execute the user choice wether to merge ,add or replace the data

ii. First the project which represent the button **<u>Synchronize</u>** which consist of :

1. **Class "SynchronizeFileMaker":**
    - **a.** First gets the revit document
    - **b.** Then check if there is no smart schedule created previously
    - **c.** Then give the user the option wheter to proceed and create a blank smart schedule or stop
    - **d.** Then the class will connect to the file maker server and extract the data an but them in a list using the class "**ExtractDataFromSchedule**"
    - **e.** Then it will call the WPF named "**MainWindow**" which will give the user the choice wether to extract data or update the data or create new data and execute the required action an retrieve the server data to be used later
    - **f.** Then it will call a method named "Identify" to call the class named "**IdentifyingChosenMethod**" Which will update the smart schedule in the case of extracting the data with the extracted data or in the case of creating new items and extract the data and update the smart schedule with the extracted data
2. **WPF** "**MainWindow**":
    - **a.** The main event access the filemaker server
    - **b.** The Create event create the new items using method **"CreateButton_Copy_Click"** then extract the data using the method "**ExtractingNewItems**" after creating the new items
    - **c.** The Extract event only Extract the data from the server using the method "**Extract_Click**"
    - **d.** The update event which only update the data in the server using the data sent from the main window