Revit Plugin Development Design Document

❖Title & Overview

- Title: **Asite Revit Plugin**
- Overview: Asite an open, cloud-based platform makes it easier for users to plan, design & seamlessly share the information across the entire supply chain.

❖Introduction

- Asite Revit Plugin is a collaborative platform that allows all parties working on a project to have access to the 3D models.
- Through this they can effectively communicate any information across the project in real-time, like if an issue needs attention, creating an issue in Issue manager with priority.

❖Functional Requirement

- Features:
    - Communicate and track issues across BIM authoring platforms, including between different versions of Revit
    - Create issue pinpoints from any given location in your 3D models
- Use Case:
    - User can modify & update the document
    - User can identify the issues and add them in Issue Manager with priority
    - Can upload the modified & updated file to Asite CDE
    - Can export rvt file to nwc/ifc/dwfx file and upload them as secondary file so that it can open file in Navisworks in less time

❖Technology Stack

- Programming Language: C#

- Frameworks/Libraries: .NET Framework 4.8.1
- APIs/Interfaces:
  - Autodesk Revit APIs
  - Autodesk Windows Library (AdWindows)
  - Newtonsoft.Json
  - RestSharp
  - Microsoft.Web.WebView2

❖ Asite CDE

- Common Data Environment (CDE) is a digital information platform that centralizes project data storage and access, typically related to a construction project and building information modeling (BIM) workflows.
- Asite CDE is an online platform that provides a centralized location to store, access, share, and collaborate on all documents related to a project, giving you visibility into your project's risks ahead of time.
- Asite Common Data Environment (CDE) controls your data and enables BIM compliance.

❖ Project Setup Guide

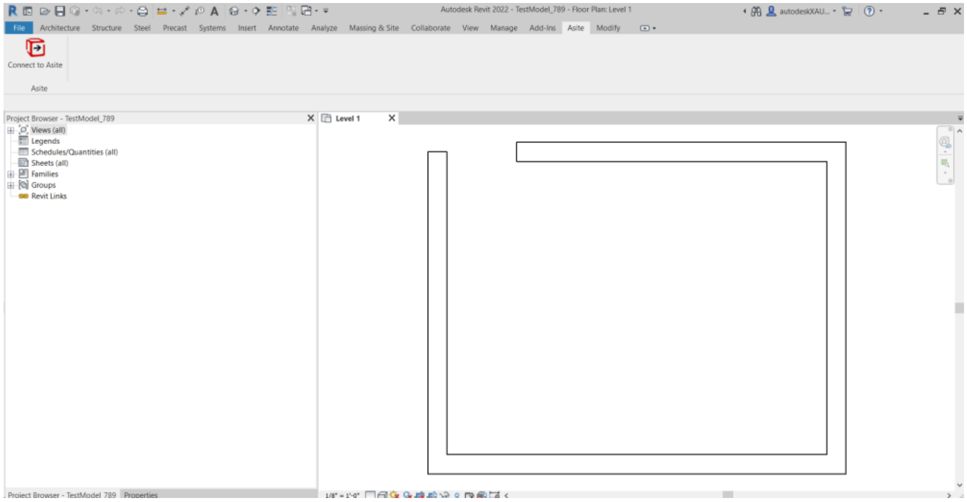
- Install Visual Studio 2017/2019/latest & Autodesk Revit
- Clone and pull the bitbucket repo of plugin: git@bitbucket.org:asitesol/bcf-plugins.git
- Open the solution in visual studio in Administrator mode
- Add the required packages from Manage NuGet
- Revit Plugin has support for Revit versions from 2019 to 2024.
- Check the version of Revit installed on your system and accordingly change the build path
  - Right click on project, open the properties, under build change the output path to: C:\Users\UserName\AppData\Roaming\Asite Solutions Ltd\Revit+RevitVersion(eg:Revit2022)\

- Make sure to add the RevitAsiteBCFPlugin.addin file in the Addin folder of Revit
- User can now build the project/solution
- Build the project RevitAsiteBCFPlugin from the Solution Explorer, until successful builds.

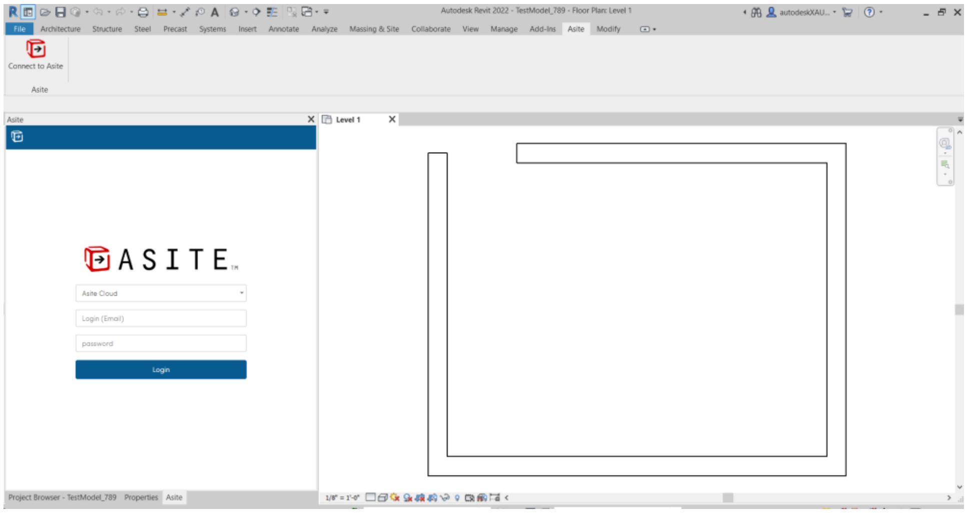
❖**Plugin Login Component**

- Launch the Revit application after successful build of project.

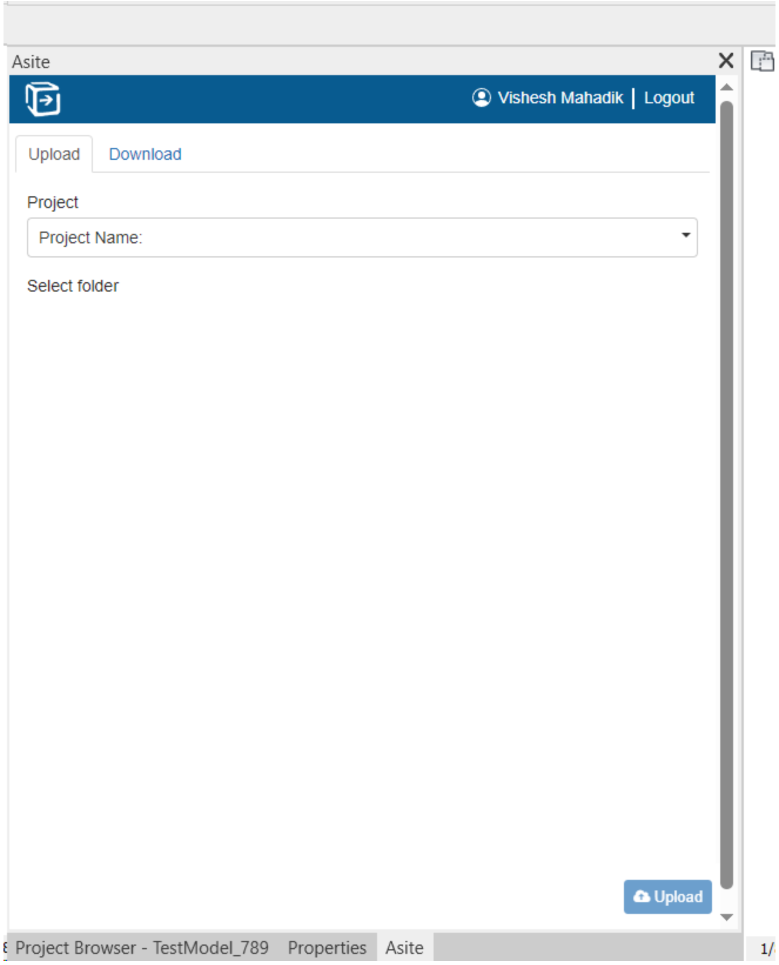
- Check the Asite Ribbon tab which include the Connect to Asite button as below:



- Clicking on Connect to Asite loads the Login Component (Login web Page) as below:



- Clicking on Connect to Login gives the call to doLogin() fucntion from MainPage.xaml.cs file in solution explorer.
- Cloud data is the platform on which user is working on: QA/STG/SB/LIVE(Passed through doLogin() function from MainPage.xaml.cs
- Provide the valid credentials: UserName & Password based on the platform
- Validated credentials redirects to onLoginSuccess() function, response from AsiteCDE with cloud details, Stored File User Id & SessionId.
- After validating the credentials, the user is redirected to upload page with download tab as below:

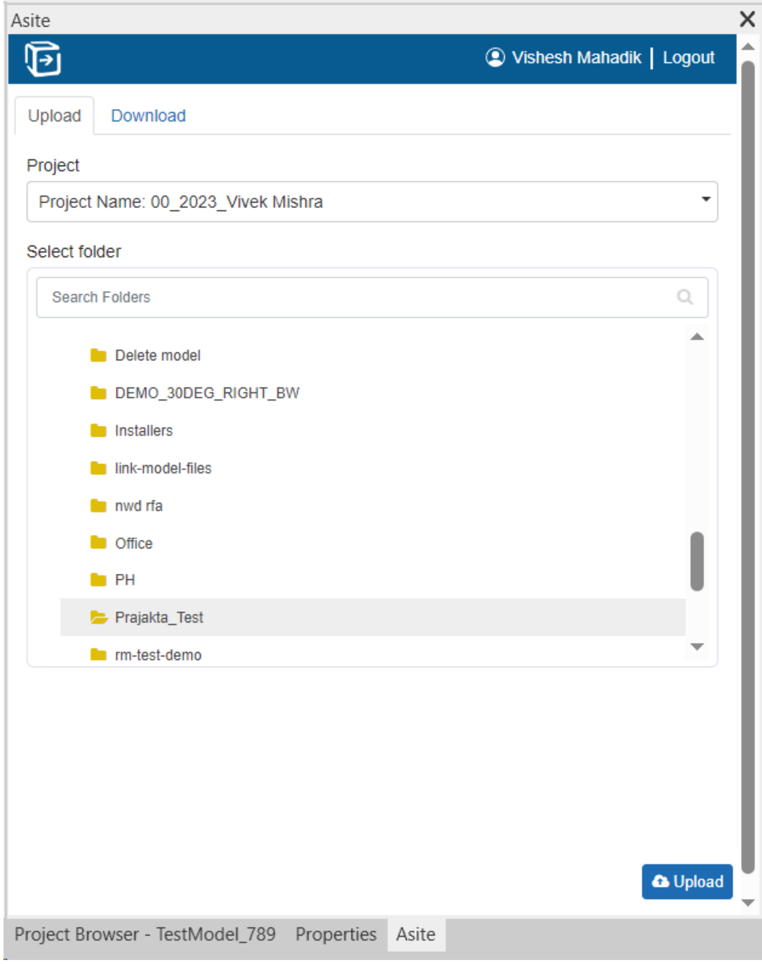


❖**Plugin Components**

1) Upload Component

a. Upload Page includes the project name & folder name, selecting project name enables folder selection, which includes list of all folder & subfolders present under the selected project

b. Select the folder/subfolder to upload the model. Selecting folder enables the upload button



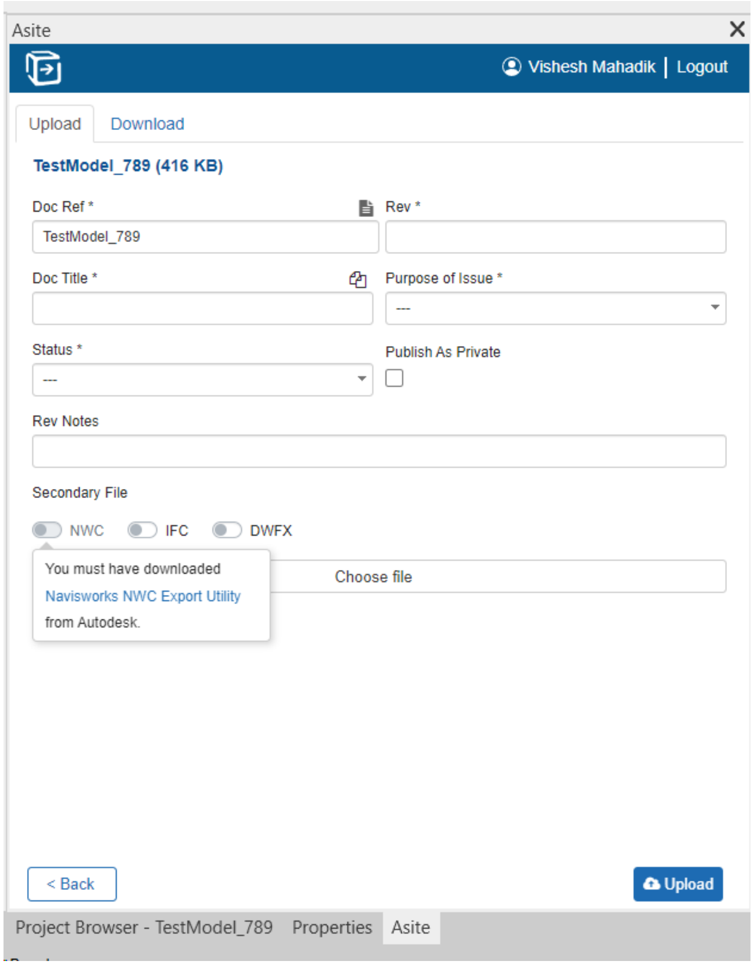
Project

Project Name: 00_2023_Vivek Mishra

Select folder

Prajakta_Test

c. Clicking on upload button redirects to upload metadata page, which includes Doc ref (Name of rvt file), revision number, Document title, etc.



TestModel_789 (416 KB)

Doc Ref * TestModel_789

Rev *

Doc Title *

Purpose of Issue * ---

Status * ---

Publish As Private □

Rev Notes

Secondary File

NWC IFC DWFX

You must have downloaded Navisworks NWC Export Utility from Autodesk.

Choose file

d. The active file is considered to be uploaded as Primary file. With primary file we can upload the secondary file nwc/ifc/dwfx/all three. To export nwc user needs to install Navisworks NWC export utility.

   i. NWC Export Use case: Upload heavy files from Revit Application and synching & opening the same file in Navisworks takes time. If we attach the nwc as secondary file, the nwc file is synched & opened in Navisworks instead of rvt file.

e. If user selects only 1 secondary file it is uploaded as single secondary file, but if user selects more than 1 secondary file then that files are exported as a zip file and that zip is uploaded as secondary file. Exporting of secondary file is handled in ExportUploadSecondaryFile() function in MainPage.xaml.cs file

f. If a user wants to upload any other format file from the local system, it can be uploaded by clicking on Choose file button which redirects to folder explorer.

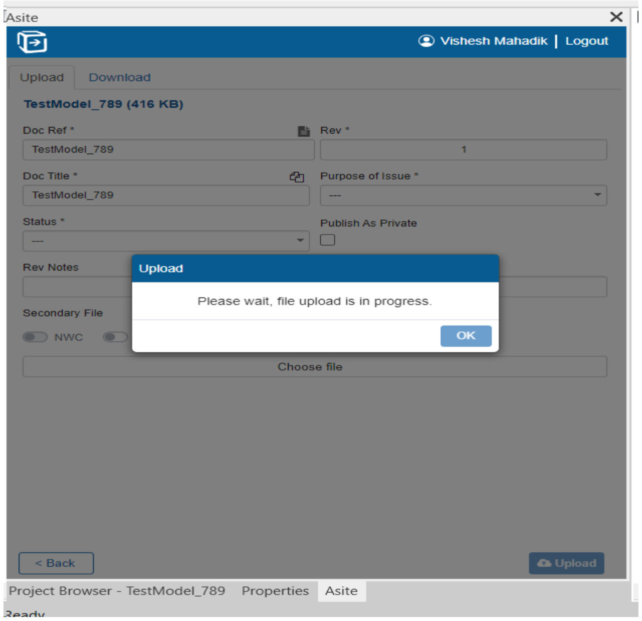
g. Clicking on the Back button redirects the user back to the project listing page.

h. Clicking on the Upload button calls uploadAttributesData() function from MainPage.xaml.cs file. Through this function uploadFiles() function is called with ExternalEvent class.

i. UploadFiles() function includes the call to all the functions that are required to upload the primary file.

j. After successful upload of primary file the secondary file is upload if set as true.

k. To Export & upload the type property json or not is set in setRevitUploadWithoutTypePropertyJsonValue() function response coming from Asite CDE



TestModel_789 (416 KB)

Doc Ref * TestModel_789

Rev * 1

Doc Title * TestModel_789

Purpose of Issue * ---

Status * ---

Publish As Private ☐

Rev Notes

Secondary File

NWC

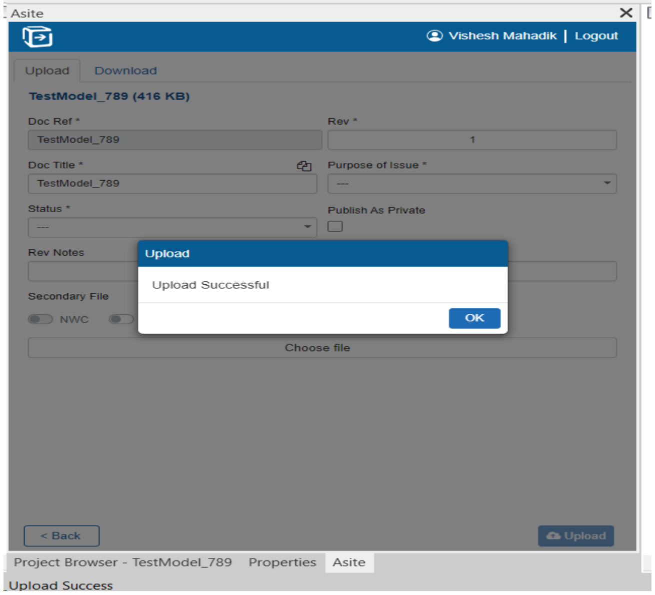
Upload

Please wait, file upload is in progress.

Ready

l. Uploading ongoing process is shown through the above UI.

m. Successful upload gives the call to uploadSuccess() function



TestModel_789 (416 KB)

Doc Ref * TestModel_789

Rev * 1

Doc Title * TestModel_789

Purpose of Issue * ---

Status * ---

Publish As Private ☐

Rev Notes

Secondary File

NWC

Upload

Upload Successful

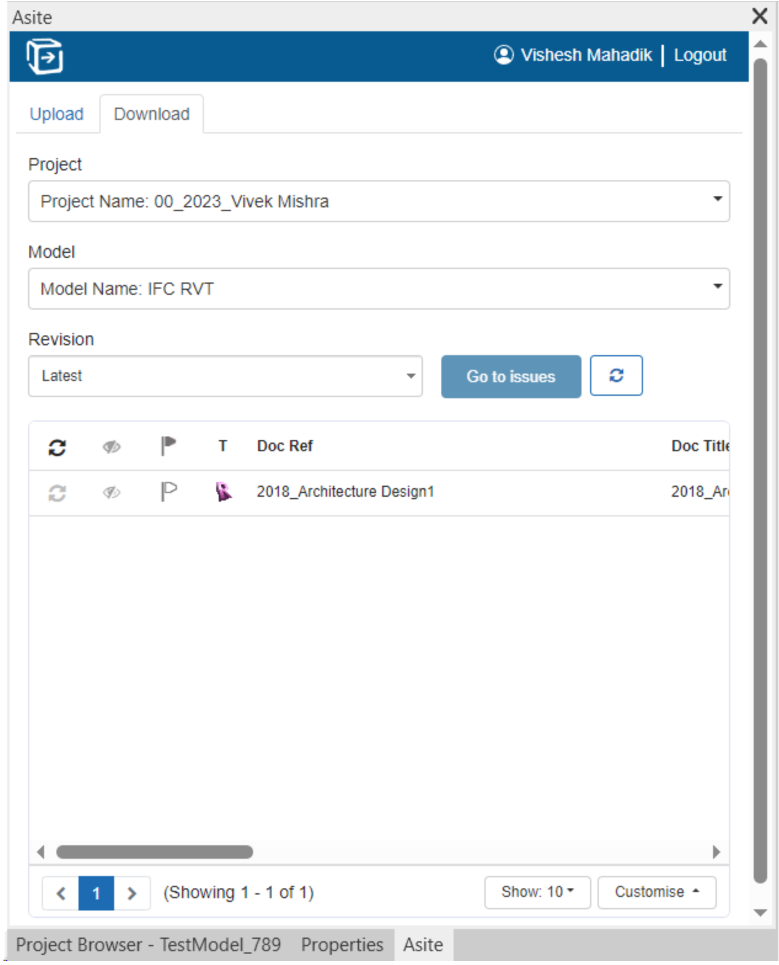
Upload Success

n. Upload Failure gives the call to uploadFailure() function.

2) <u>Download Component</u>

a. The Download Page includes the first selection of Project that enables selection of model from the list.

b. Selection of model from list displays the list of all rvt & rfa files with sync & view icon



Project: Project Name: 00_2023_Vivek Mishra

Model: Model Name: IFC RVT

Revision: Latest

T	Doc Ref	Doc Title
	2018_Architecture Design1	2018_Ar

(Showing 1 - 1 of 1)

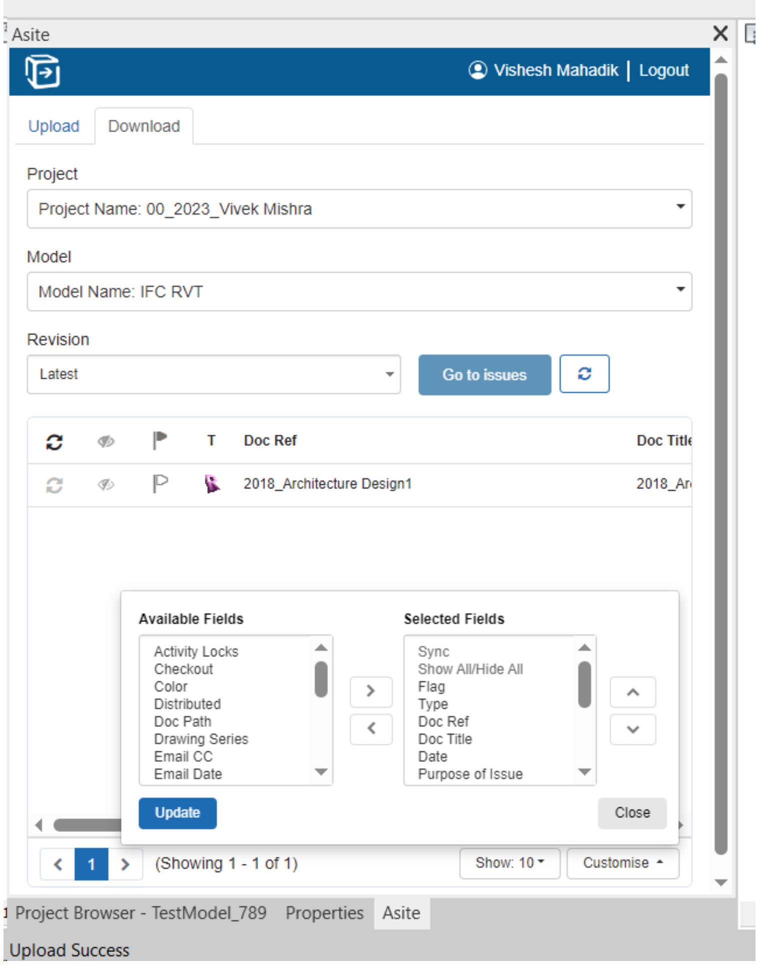
c. While loading the list of files of model the getDownloadStatus() function is called to check the state of file (already synced or not). If already synced the grey sync icon turns green and the eye icon is enabled to operate.

d. If not synced, clicking on sync icon gives a call to startDownload() which includes call back event of DownloadFileCompleted(). User can stop ongoing download of file by clicking on grey sync icon which calls stopDownload() function.

e. Once the download is completed the grey sync icon turns green and the eye icon is enabled to operate.

f. Clicking on the eye icon opens the file which is handled in openFiles() function & we can close the opened file by clicking on same opened eye icon which is handled in closeFiles() function.

g. Customizing the columns within the file listing can be done using the customize dropdown as below:



Project: Project Name: 00_2023_Vivek Mishra

Model: Model Name: IFC RVT

Revision: Latest

T	Doc Ref	Doc Title
	2018_Architecture Design1	2018_Ar

Available Fields: Activity Locks, Checkout, Color, Distributed, Doc Path, Drawing Series, Email CC, Email Date

Selected Fields: Sync, Show All/Hide All, Flag, Type, Doc Ref, Doc Title, Date, Purpose of Issue

(Showing 1 - 1 of 1) Show: 10

Upload Success

Note: Opening of lower version file in higher version of Revit starts the upgrade process

3) Issue Manager

a. Go to issues:

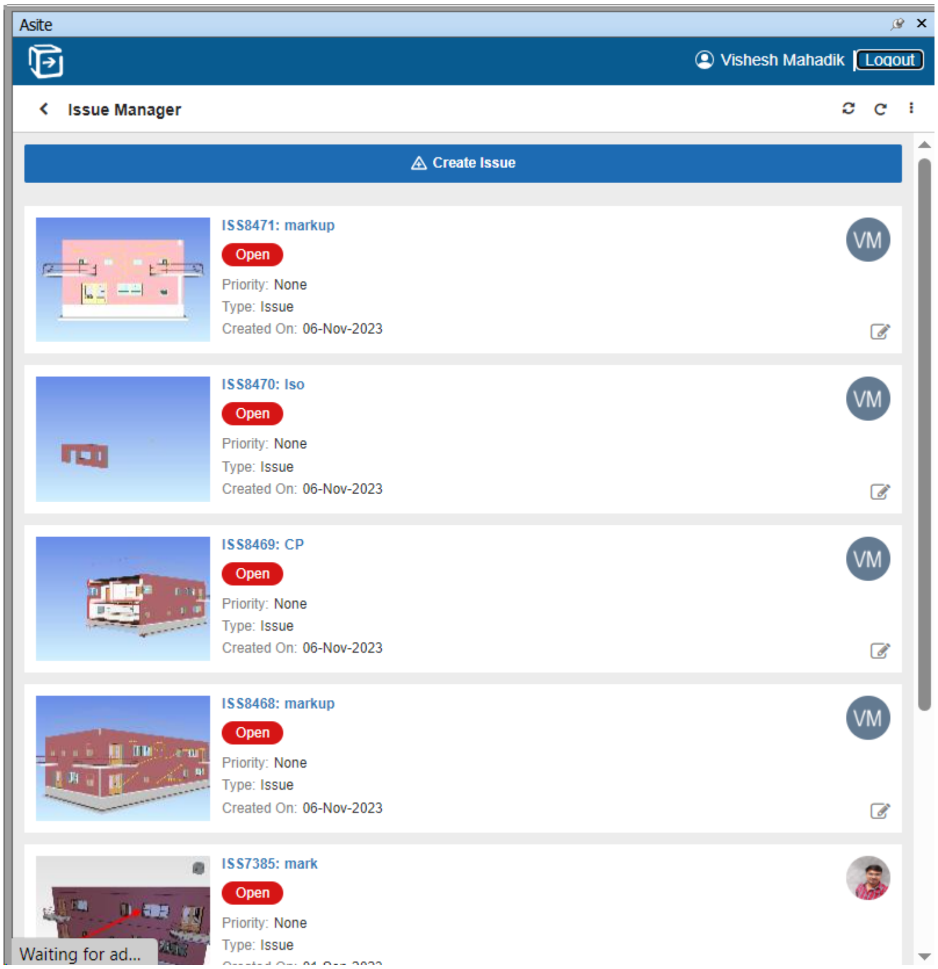
i. This button is enabled only if atleast 1 model is opened in viewer

ii. Clicking on this button opens the list of issues in Issue Manager with Create issue button, refresh button

b. Create Issue:

   i. Clicking on create issue calls CaptureView() function from MainPage.xaml.cs file and captures the viewpoint data like clipping plane, color, transparent, focus, isolate.

c. **View Issue:**



ISS8471: markup
Open
Priority: None
Type: Issue
Created On: 06-Nov-2023

ISS8470: Iso
Open
Priority: None
Type: Issue
Created On: 06-Nov-2023

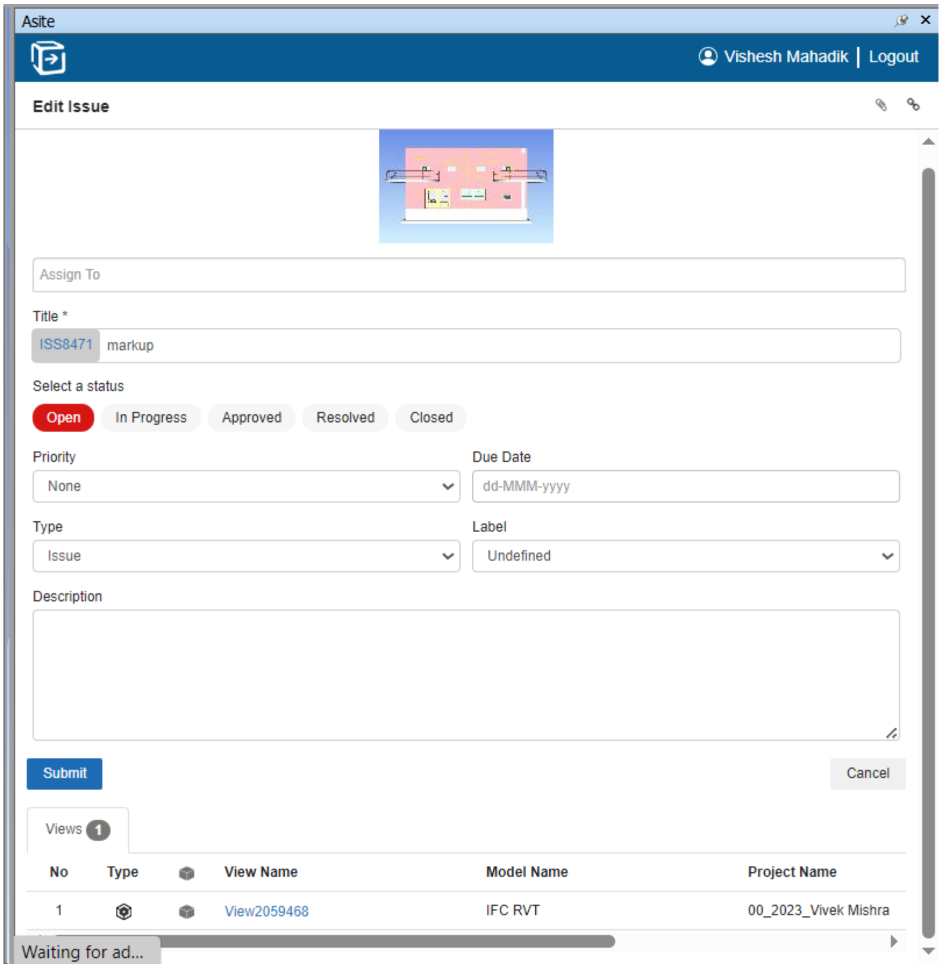
ISS8469: CP
Open
Priority: None
Type: Issue
Created On: 06-Nov-2023

ISS8468: markup
Open
Priority: None
Type: Issue
Created On: 06-Nov-2023

ISS7385: mark
Open
Priority: None
Type: Issue

   i. <u>Thumbnail:</u> Clicking on thumbnail image of issue loads the issue in the viewer and calls the RenderView() from MainPage.xaml.cs file

   ii. <u>Edit Issue:</u> Clicking on Edit issue icon loads the issue in viewer and opens the metadata of the issue, which when updated captures the new the View as link.



Asite

Vishesh Mahadik | Logout

Edit Issue

Assign To

Title *

ISS8471 markup

Select a status

[✓ Open] In Progress Approved Resolved Closed

Priority: None

Due Date: dd-MMM-yyyy

Type: Issue

Label: Undefined

Description

No	Type		View Name	Model Name	Project Name
1			View2059468	IFC RVT	00_2023_Vivek Mishra

Waiting for ad...

d. **Reset Issue to default View**: Reset Visibility Button reset the model to default visibility, and this is handled in ResetToDefaultState() function in MainPage.xaml.cs file