Project Documentation: Cesium Certification Project

Developer Name: Zainal Safwan Bin Zainal Abidin

Overview

This project is designed to meet the requirements for becoming a Cesium-certified developer, as outlined in the Cesium Certification Requirements. The application demonstrates proficiency in CesiumJS by integrating core functionalities, including KML data visualization, interactive asset management, and advanced user interactions.

Project Goal

The primary goal of this project is to showcase a practical and interactive geospatial application using CesiumJS, emphasizing:

- Loading and visualizing KML files with multiple data points.
- Providing detailed asset information upon user interaction.
- Enabling asset management through table and map-based operations.
- Ensuring seamless synchronization of data between map visuals and user interfaces.

Steps Involved

1. Loading KML Data

- A KML file containing asset points was loaded into the Cesium application.
- Each point represents an asset and is visualized on the map using custom icons.

2. Creating Asset Info Box

- When a user clicks on an asset icon, an info box appears.
- The info box provides detailed information about the selected asset, including the critical attribute: **Asset Condition** (Good, Medium, Poor, Critical).

3. Adding Asset Management Table

- A button is included in the UI to open a modal containing a table.
- The table lists all assets along with their corresponding information.
- User are able to find asset they are looking for at ease using the search filter provided at the top of the table.
- Each row includes two action buttons:
    - **Fly To:** Moves the camera to the specific asset location on the map.
    - **Edit:** Opens a form for editing the selected asset's condition.

4. Implementing Editing Functionality

- The edit form allows users to update the condition of the asset.
- Once the condition is updated:
    - Changes are immediately reflected in the asset info box on the map.
    - The table data is dynamically updated to reflect the latest changes.

Final Result

The final application successfully:

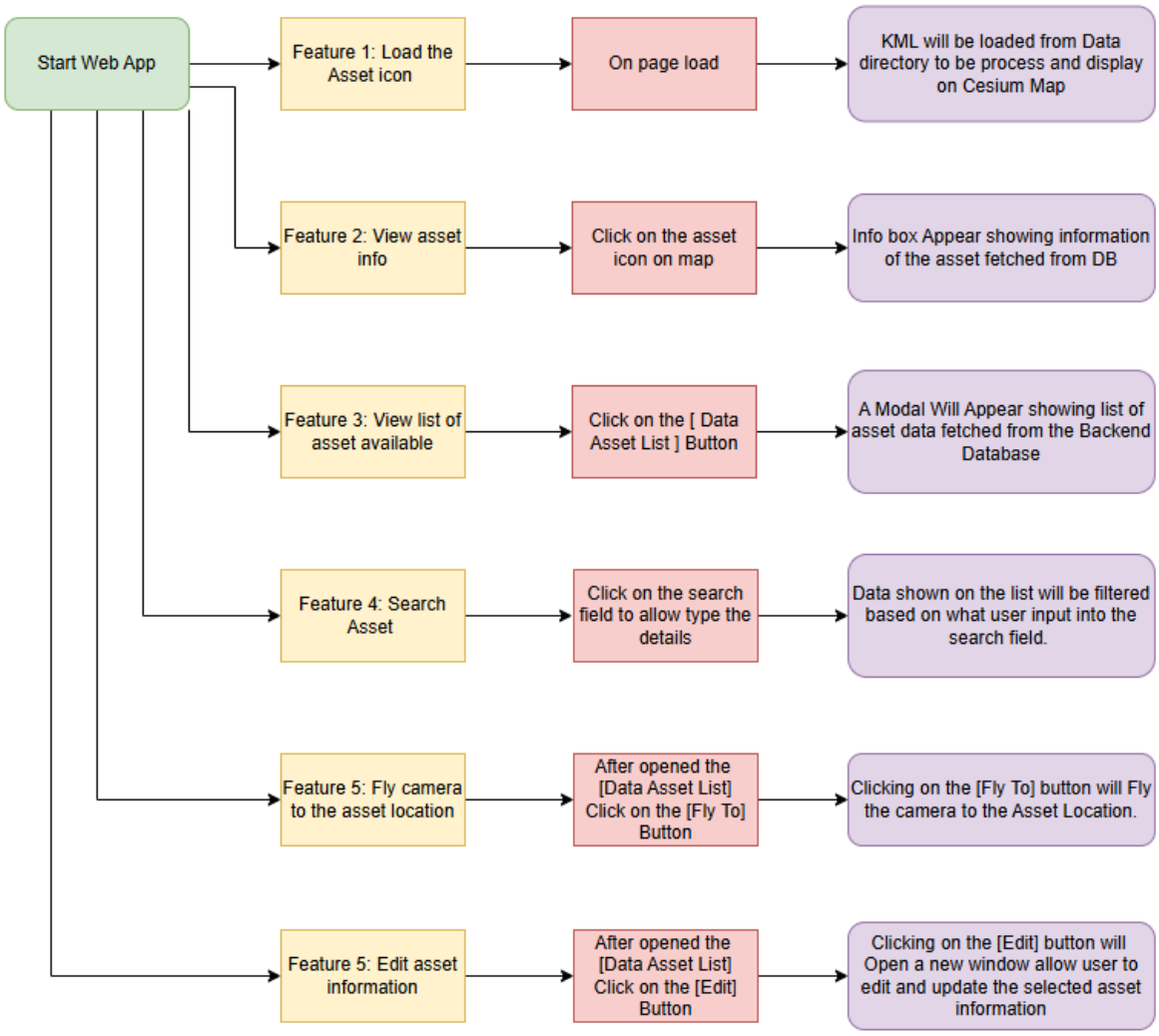
- Loads and visualizes KML data.
- Provides an interactive map where users can view asset details and manage them through a comprehensive table interface.
- Allows seamless navigation and editing of asset data, enhancing user interaction and usability.

Next Steps / Future Work

1. **Enhanced Data Visualization**
    - Integrate color-coded icons to reflect the condition of each asset directly on the map.
2. **Performance Optimization**
    - In the future more data will be added to the system, thus optimization may needed to ensure the smoothness of the system.
3. **Extended Asset Management Features**
    - Allow bulk editing of asset conditions via the table interface.
    - Allow user to redirect the app to the Google Map to allow them navigate their route to the asset selected.
    - Allow user to upload different type of asset data.
4. **Improved User Experience**
    - Include detailed tooltips and user guidance for first-time users.
    - Add animations for transitions like flying to asset points or opening modals.

By implementing these future enhancements, the application can evolve into a robust geospatial asset management platform, showcasing a deeper understanding and mastery of CesiumJS.

Diagrams



Technical Documentation

Feature 1 Code:

```
32          viewer.dataSources.add(Cesium.KmlDataSource.load('Data/Div_Culvert_Kuching.kml', options))
33              .then(function(dataSource) {
34                  const entities = dataSource.entities.values;
35
36                  for (let i = 0; i < entities.length; i++) {
37                      const entity = entities[i];
38
39                      if (entity.billboard) {
40                          entity.billboard.image = 'image/culvert.png'; // Set the new icon path
41                          entity.billboard.scale = 1.5; // Adjust the scale (make it bigger)
42                          entity.billboard.color = Cesium.Color.GRAY; //set color of the asset
43                      }
44                  }
45
46                  viewer.flyTo(dataSource);
47              })
```

Feature 2 Code:

```
function fetchAssetData(entity,assetId){
    event.preventDefault();
    $.ajax({
        url: "php/fetchData.php",
        type: "POST",
        dataType: "JSON",
        data: {
            functionName: "fetchAssetData",
            assetId: assetId
        },
        success: (data) => {
            if (data.status === 'Success') {

                showCustomPopup(entity, data.asset_data)

            } else {
                window.alert("Record not available.")
            }
        },

    });

}

function showCustomPopup(entity, data) {
    var amendRow = `<tr><th>Condition</th><td> ${data.asset_condition || 'No data available.'}</td></tr></tbody>`

    let updated_Infobox_Detail = entity._description._value.replace('</tbody></table></div>', amendRow+'</tbody></table></div>');
    $('.cesium-infoBox-description-lighter').find('table tbody').append(amendRow)

    entity._description._value = updated_Infobox_Detail
}
```

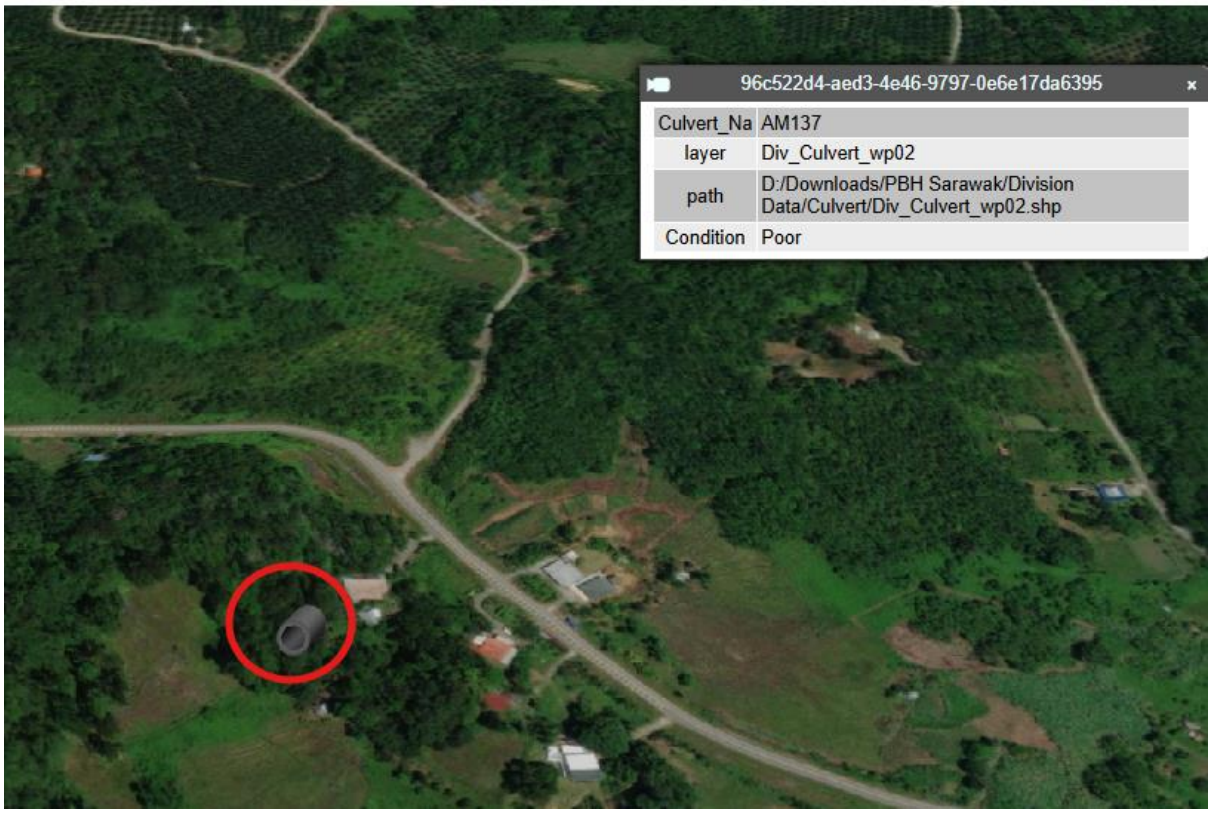
Challenges

Injecting custom info was not successful at first because the JS wasn't able to alter the table after it popup. After studying the codes and the data from the entity, it appear that the table was display after the API calling is completed, thus the easiest way to display the custom data column into the table is by modifying the value of the _description from the entity selected.

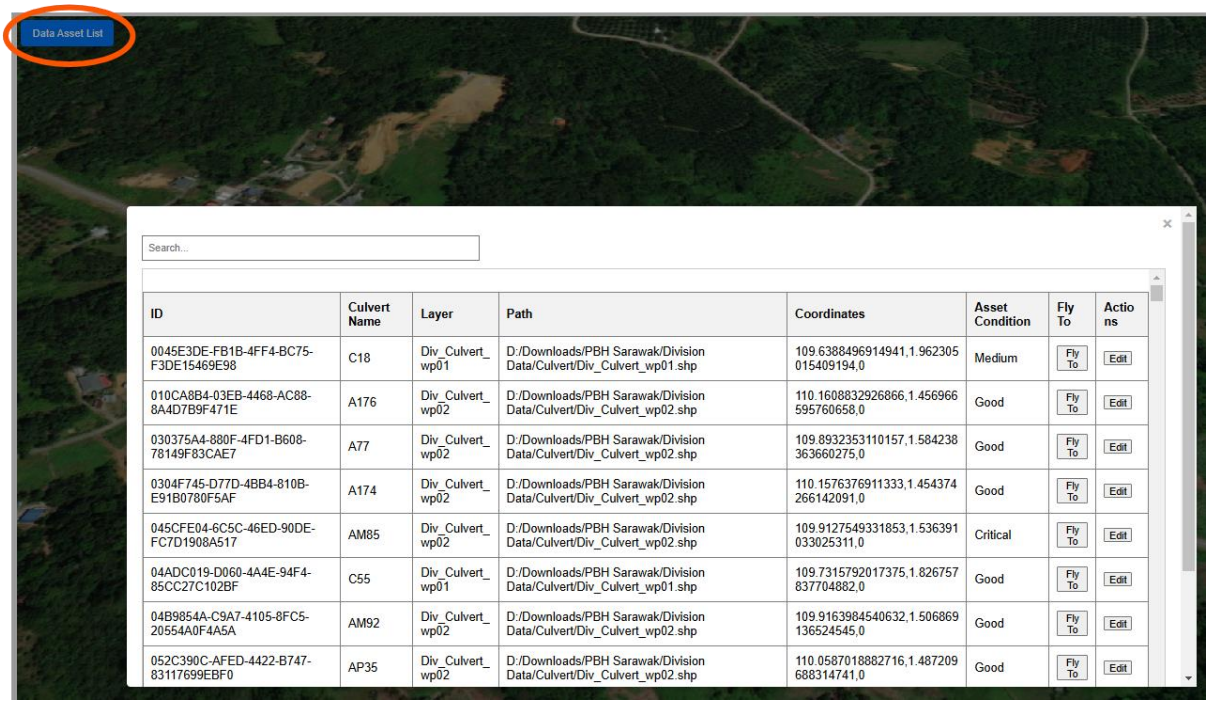
Project implementation explanation

1)After loading the system, it will automatically load the KML Asset Data Layer and the camera will fly to the loaded KML location

2)User are able to view the asset information by clicking on the icon, then the info box will appear showing the asset information.

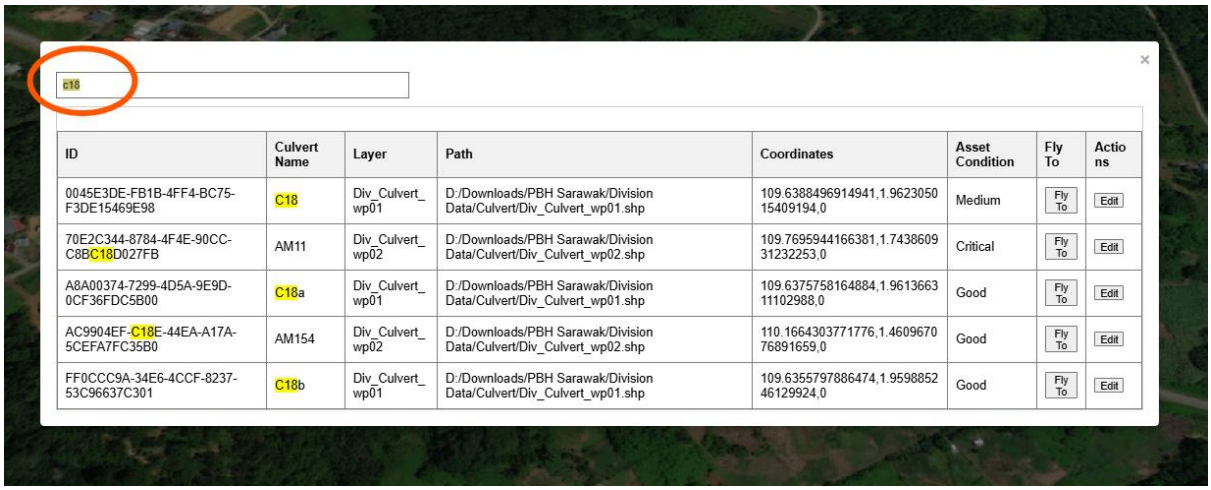


3)To view all available asset registered, user can click on the [Data Asset List] and the modal will appear displaying all asset data available.

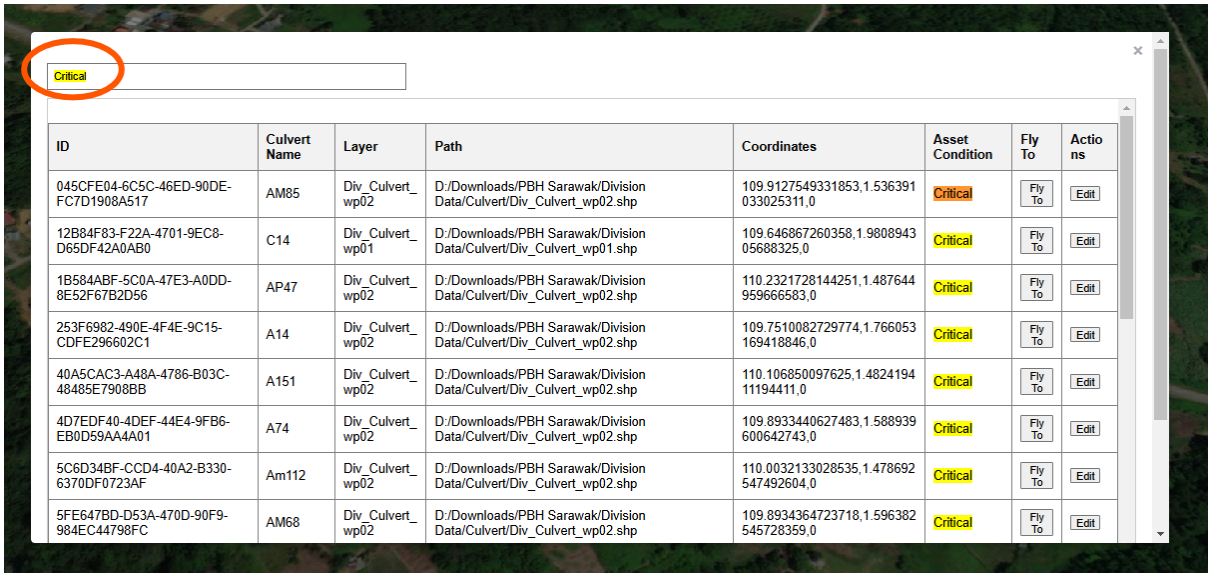


ID	Culvert Name	Layer	Path	Coordinates	Asset Condition	Fly To	Actions
0045E3DE-FB1B-4FF4-BC75-F3DE15469E98	C18	Div_Culvert_wp01	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp01.shp	109.6388496914941,1.962305015409194,0	Medium	Fly To	Edit
010CA8B4-03EB-4468-AC88-8A4D7B9F471E	A176	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	110.1608832926866,1.456966595760658,0	Good	Fly To	Edit
030375A4-880F-4FD1-B608-78149F83CAE7	A77	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	109.8932353110157,1.584238363660275,0	Good	Fly To	Edit
0304F745-D77D-4BB4-810B-E91B0780F5AF	A174	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	110.1576376911333,1.454374266142091,0	Good	Fly To	Edit
045CFE04-6C5C-46ED-90DE-FC7D1908A517	AM85	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	109.9127549331853,1.536391033025311,0	Critical	Fly To	Edit
04ADC019-D060-4A4E-94F4-85CC27C102BF	C55	Div_Culvert_wp01	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp01.shp	109.7315792017375,1.826757837704882,0	Good	Fly To	Edit
04B9854A-C9A7-4105-8FC5-20554A0F4A5A	AM92	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	109.9163984540632,1.506869136524545,0	Good	Fly To	Edit
052C390C-AFED-4422-B747-83117699EBF0	AP35	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	110.0587018882716,1.487209688314741,0	Good	Fly To	Edit

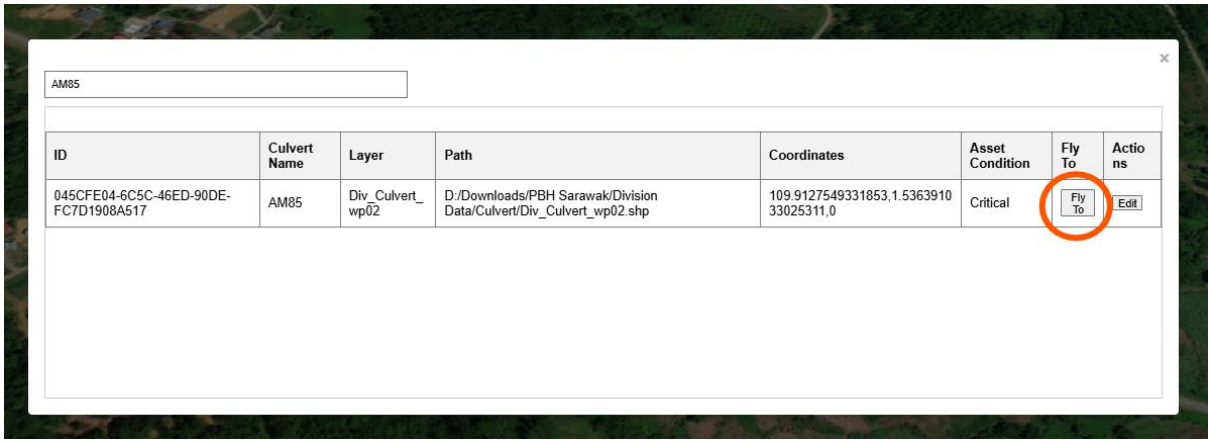
4) To ease user for searching an asset, search filter feature is provided, user can easily filter the data they need by using the search fields provided



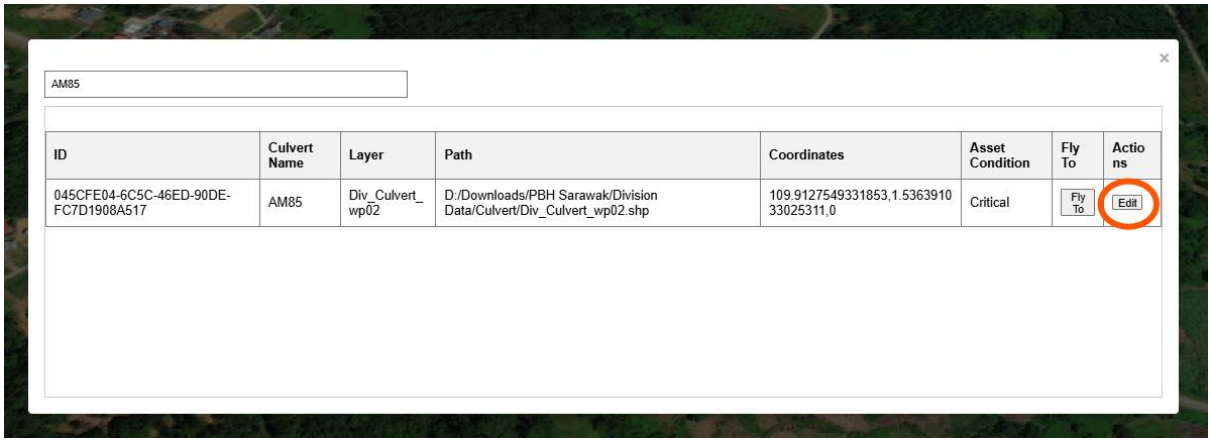
ID	Culvert Name	Layer	Path	Coordinates	Asset Condition	Fly To	Actions
0045E3DE-FB1B-4FF4-BC75-F3DE15469E98	C18	Div_Culvert_wp01	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp01.shp	109.6388496914941,1.962305015409194,0	Medium	Fly To	Edit
70E2C344-8784-4F4E-90CC-C8BC18D027FB	AM11	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	109.7695944166381,1.743860931232253,0	Critical	Fly To	Edit
A8A00374-7299-4D5A-9E9D-0CF36FDC5B00	C18a	Div_Culvert_wp01	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp01.shp	109.6375758164884,1.961366311102988,0	Good	Fly To	Edit
AC9904EF-C18E-44EA-A17A-5CEFA7FC35B0	AM154	Div_Culvert_wp02	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp02.shp	110.1664303771776,1.460967076891659,0	Good	Fly To	Edit
FF0CCC9A-34E6-4CCF-8237-53C96637C301	C18b	Div_Culvert_wp01	D:/Downloads/PBH Sarawak/Division Data/Culvert/Div_Culvert_wp01.shp	109.6355797886474,1.959885246129924,0	Good	Fly To	Edit



5) While looking for the asset 1 by 1 may took long time to find out which one is the asset icon, user are able to use the Fly To feature. Clicking on the Fly To button will move the camera to the location of the selected asset data.



6) Asset Data information may need some update from time to time to ensure the project or the flow of the work done on the site is smooth and safe. User are able to perform this by clicking on the [Edit] button on the Data Asset List and submit the changes to update the information directly into the Database.



A form will appear and user can update the information onto the database by submitting this form.