FlightGear App
Main clases and app architecture:

App Usage Flow:

Run the application. In the mainWindow there are instruction for download and set up the FlightGear simulator. After setting up FlightGear, there is a "settings" button for start using the app. When clicking on the "settings" button, the settings window will open. On settings window, the user will have to give the app 3 full path of 3 files: XML file - which define the names of the features and the data which the user wants to investigate (which also required to be in the particular folder in FlightGear simulator). CSV file –a file which contains data and information about the normal flight. Second CSV file – a file which contains data and information about the flight which the user wants to investigate. After uploading those 3 files, the app is learning the normal flight, processing the flight data that the user wishes to investigate and ready for user use.

The app is built from several main clases. The first two clases are the basic capabilities of the app. The other clases works on there own with a connection to the app through the app window class. The main clases are:

MianWindow:
Instruction for installing and setting up FlightGear simulator before using the app.

SettingsWindow:
Uploading 3 files before start using the app, as described above.

AppWindow:
The app main window. Connecting all the features and responsible for transferring data between clases if required (e.g. – current time).
Display a button for uploading a new anomaly flight csv file.

FilesLoaderMVVM:
Parsing and processing the uploaded files using IFileParser interface.

TimeControllerMVVM:
Control video time with time slider, playback speed and 3 button – play, pause, start over.

WheelMVVM:
Display a joystick which shows live image of the values of 2 features: aileron and elevator.
Display sliders which shows live image of 2 other features: rudder and throttle.

FilesInfoMVVM:
Display 3 clocks which shows live image of the values of 3 features: speed, direction and altitude. Display 3 other values of features on time: yaw, roll and pitch.

GraphConrollerMVVM:
Display a list of all features. The user can choose one feature and investigate it with the following graphs which are updating in real time: The selected feature graph which represent the time at X axe and the feature's values at Y axe. The most correlative feature of the selected feature graph which represent the time at X axe and the feature's values at Y

axe. The Linear Regression graph which represent the selected feature values at X axe and the most correlative feature at Y axe.

DllAlgorithmHandlerMVVM:

Display a button that when the user clicks it a LoadDllAlgorithm window is open.
Responsible to connect to the app window, pass and get relevant information.

LoadDllAlgorithm:

Window which responsible for getting a full path of dll algorithm which the user wants to use.

DllAlgorithmGraphMVVM:

Display a graph which draw by the uploaded dll.
Display a list of anomalies which the user can click on and move to the exact step of anomaly at the flight simulation.
Display a button that when the user clicks it a WriteYourOwnAlgorithmWindowwindow is open.

WriteYoureOwnAlgorithmWindow:

Display instructions for writing user's own dll anomaly detection algorithm.

Note: MVVM includes: model, view model, window/user control with xaml.cs + xaml.