```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Threading;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;

namespace File_Search
{
    class FileException : Exception
    {
        public FileException(string exception)
        {
            MessageBox.Show(exception);
        }
    }

    public partial class MainWindow : Window
    {

        Thread[] tobj;
        DriveInfo[] allDrives;
        static ListBox listbox;
        static Mutex mutex;
```

```
        public MainWindow()
        {
            InitializeComponent();
            tobj = null;
            mutex = new Mutex();
            allDrives = DriveInfo.GetDrives();
            tobj = new Thread[allDrives.Length];
            listbox = new ListBox();
            listbox.MouseDoubleClick += new
MouseButtonEventHandler(ListboxMouseDoubleCilcked);
            ResultGrid.Children.Add(listbox);

        }

        private void Listener(Object sender, RoutedEventArgs e)
        {
            String FileName = FileNameTextBox.Text;
            FileNameTextBox.Text = null;
            listbox.Items.Clear();

            Button btn = (Button)sender;

            switch (btn.Uid)
            {
                case "search":
```

```
                    if (string.IsNullOrEmpty(FileName))
                    {
                        MessageBox.Show("Please Enter FileName !");
                        return;
                    }

                    int i = 0;

                    foreach (DriveInfo drive in allDrives)
                    {
                        if (drive.DriveType.ToString().Equals("Fixed"))
                        {
                            tobj[i] = new Thread(() =>
{ DisplayText( GetFiles(drive.ToString(), FileName + "*"), drive.ToString()); });
                            tobj[i].Start();
                            i++;
                        }

                    }

                    break;

                case "reset":
                    FileNameTextBox.Text = null;
                    listbox.Items.Clear();
                    break;
            }
```

```
        }

        public static IEnumerable<String> GetFiles(String path, String pattern)
        {

            return Directory.EnumerateFiles(path,
pattern).Union(Directory.EnumerateDirectories(path).SelectMany(a =>
            {
                try
                {
                    return GetFiles(a, pattern);
                }
                catch (UnauthorizedAccessException)
                {
                    return Enumerable.Empty<String>();
                }
            }
            ));



        }

        public static void DisplayText(IEnumerable<String> Enum,String drive)
        {
            if (!Enum.Any())
            {
```

```
                Application.Current.Dispatcher.Invoke(new Action(() =>
                {

                        listbox.Items.Add("File is not availble in "+ drive +"
drive");

                }));
                return;
            }

            try
            {
                mutex.WaitOne();

                Application.Current.Dispatcher.Invoke(new Action(() =>
                {
                    foreach (String s in Enum)
                    {
                        listbox.Items.Add(s);
                    }
                }));

                mutex.ReleaseMutex();
            }
            catch (Exception e)
            {
                throw new FileException(e.Message);
            }
```

```
        }

        private  void ListboxMouseDoubleCilcked(object sender, MouseButtonEventArgs
e)
        {

            ListBox lb = (ListBox)sender;

            if (File.Exists((String)lb.SelectedItem))
            {

                Process myProcess = new Process();

                ProcessStartInfo myProcessStartInfo = new
ProcessStartInfo(lb.SelectedItem.ToString());

                myProcess.StartInfo = myProcessStartInfo;

                myProcess.Start();
            }




        }
    }
}
```