

# Kiến trúc phần mềm

## Đồ án cuối kỳ

### 1 Thông tin sinh viên

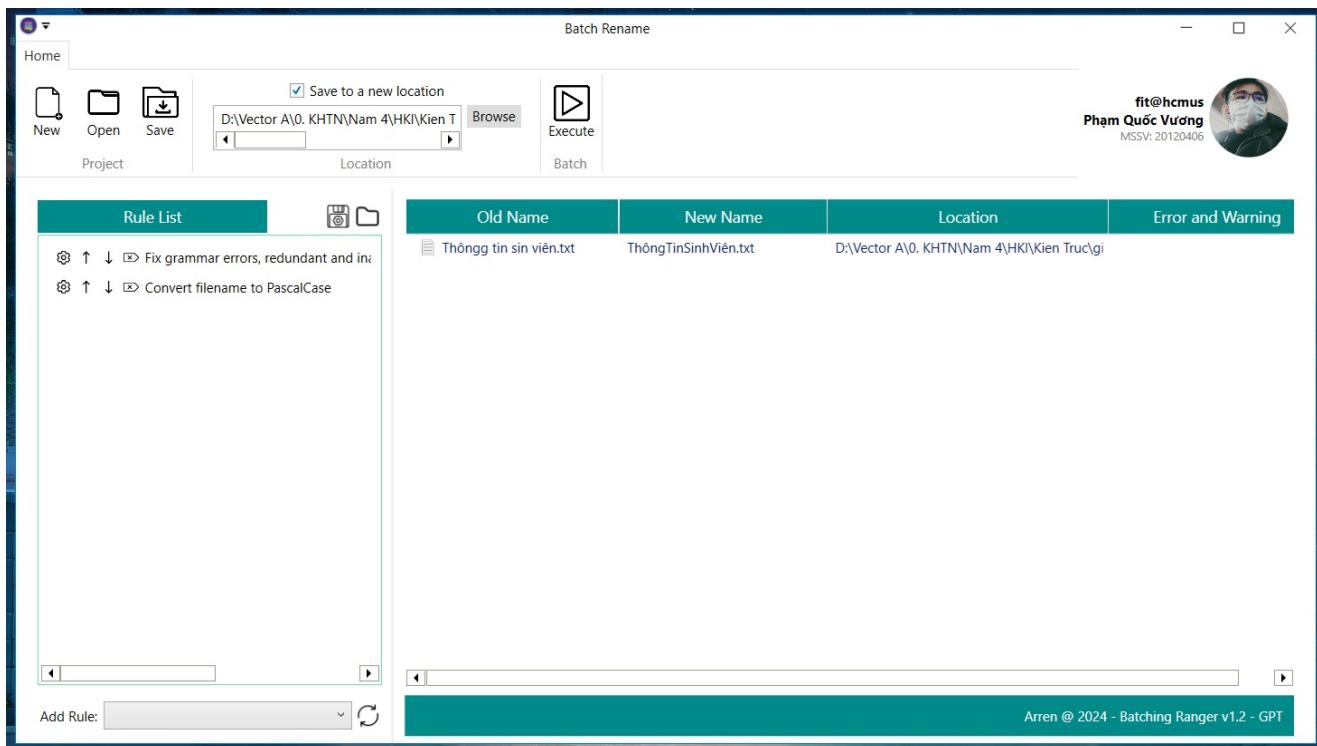
- Họ tên:** Phạm Quốc Vương
- MSSV:** 20120406
- Email:** 20120406@student.hcmus.edu.vn
- Phone/Zalo:** 0777 815 978

### 2 Khái quát về ứng dụng

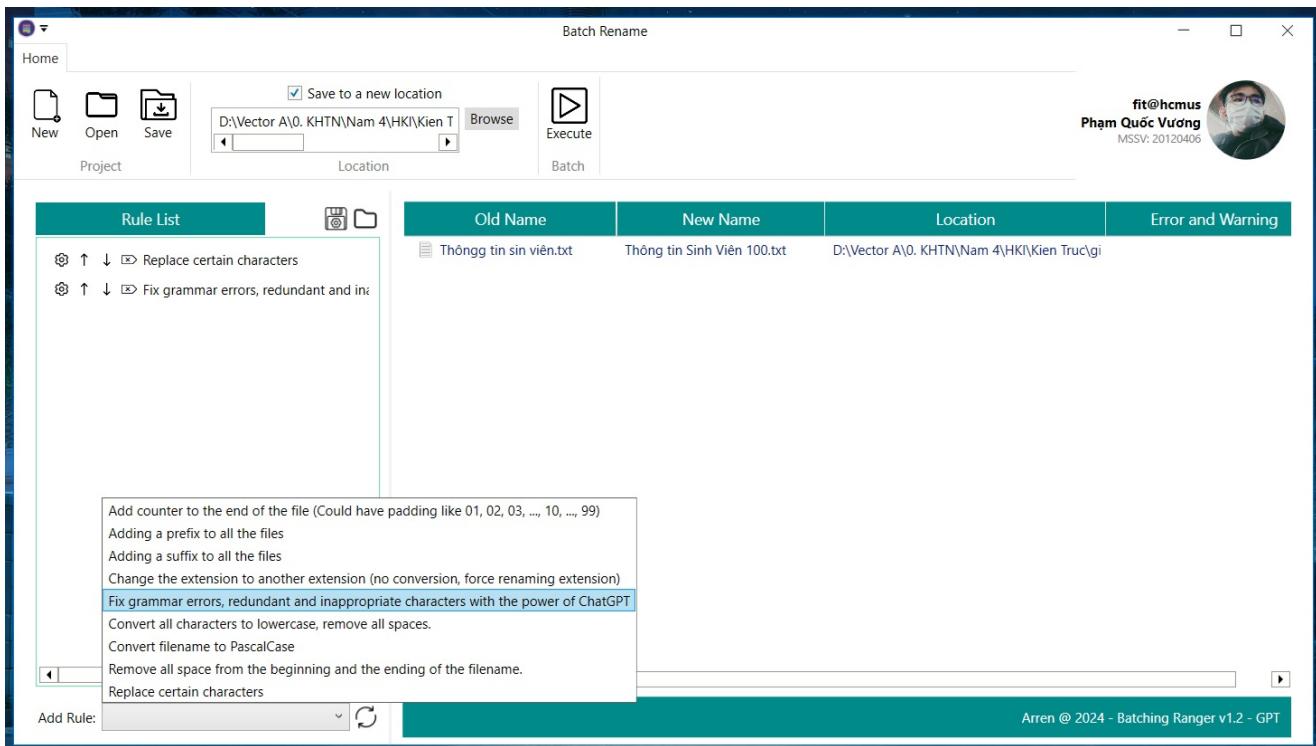
- Tên ứng dụng:** Batching Ranger v1.2 - GPT.
- Hệ điều hành hỗ trợ:** Windows 11.
- Chức năng:** đổi tên file/folder hàng loạt dựa trên danh sách luật đổi tên do người dùng chọn từ các luật (rule) mà chương trình cung cấp.
- Phiên bản ban đầu của ứng dụng:** Đồ án môn **Lập trình Windows** với tất cả các phần của hệ thống do tự tay sinh viên xây dựng/phát triển bằng rất nhiều tâm huyết (**nhóm** ở môn cũ và cả hiện tại **chỉ có 01 sinh viên**).

### 3 Ảnh chụp giao diện

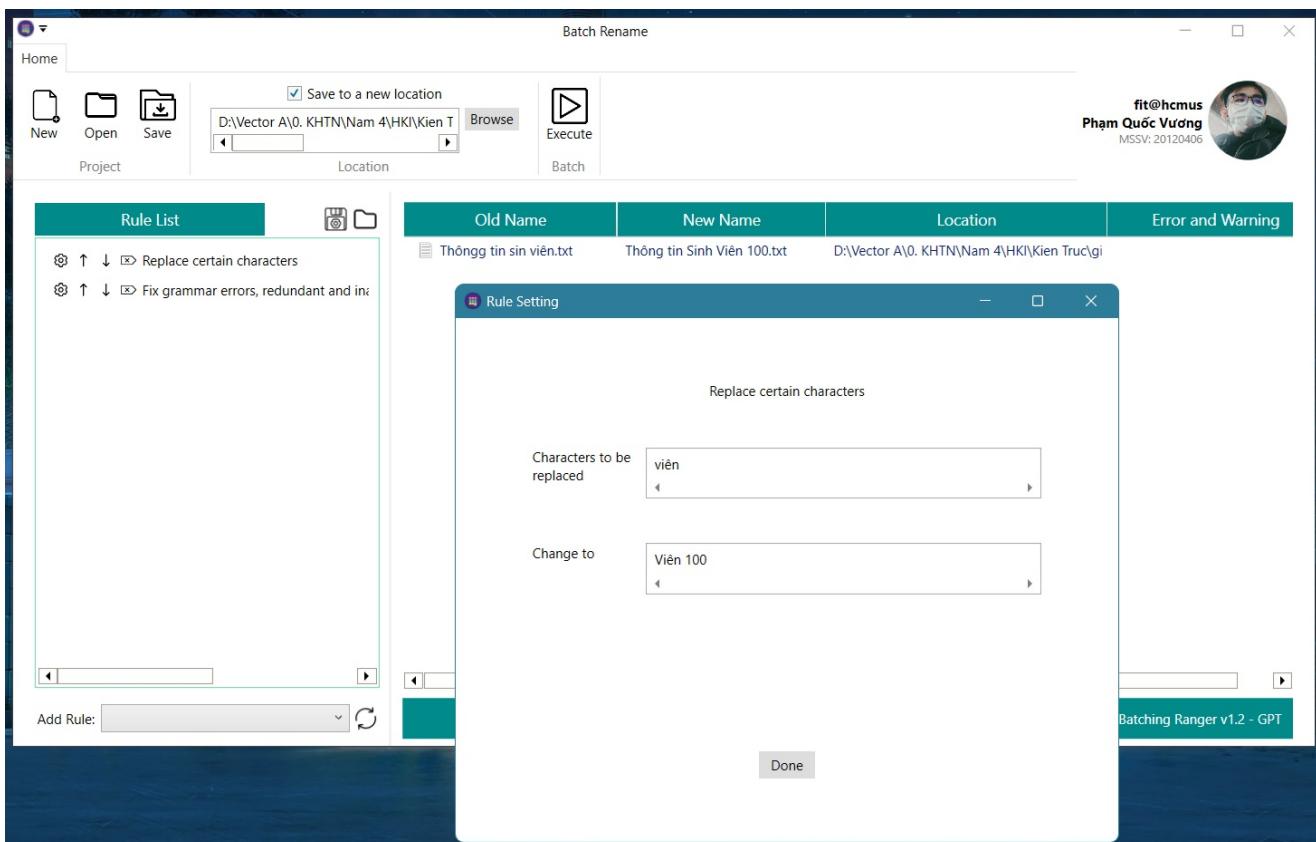
- Giao diện chính:



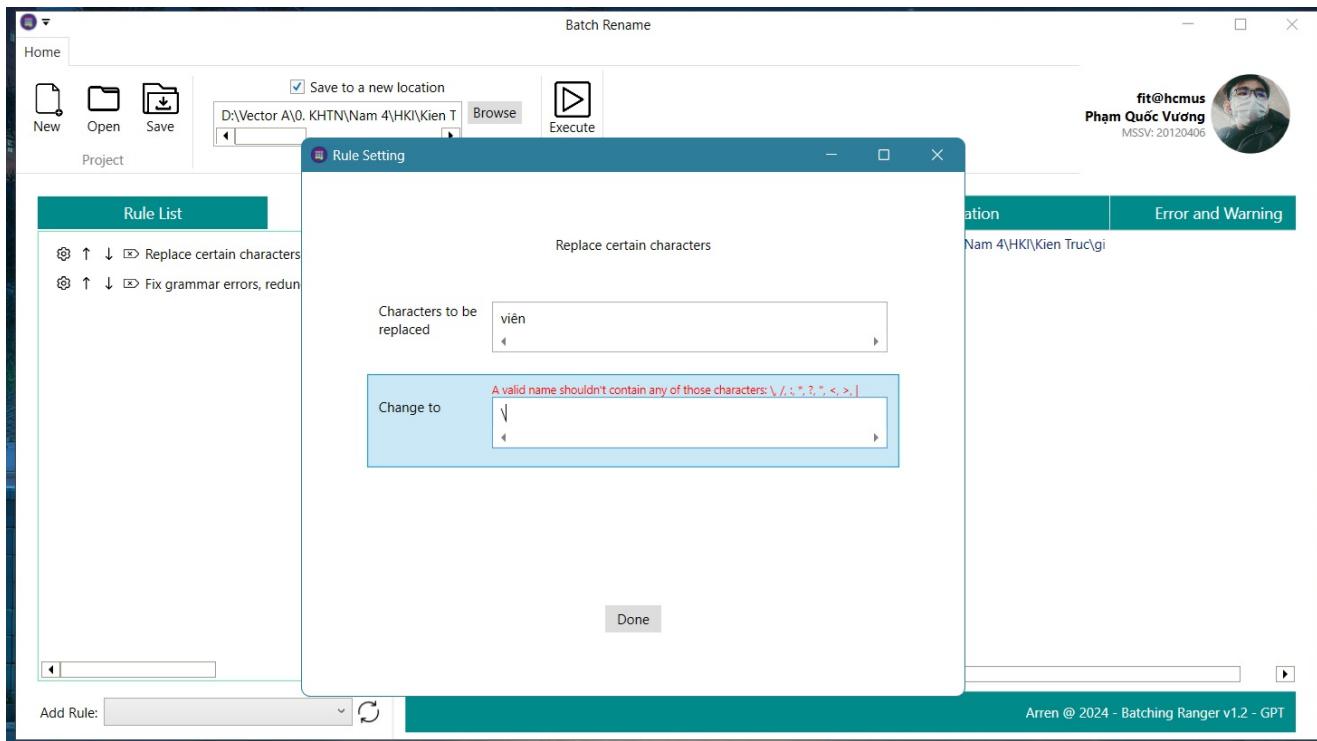
- Danh sách luật đổi tên:



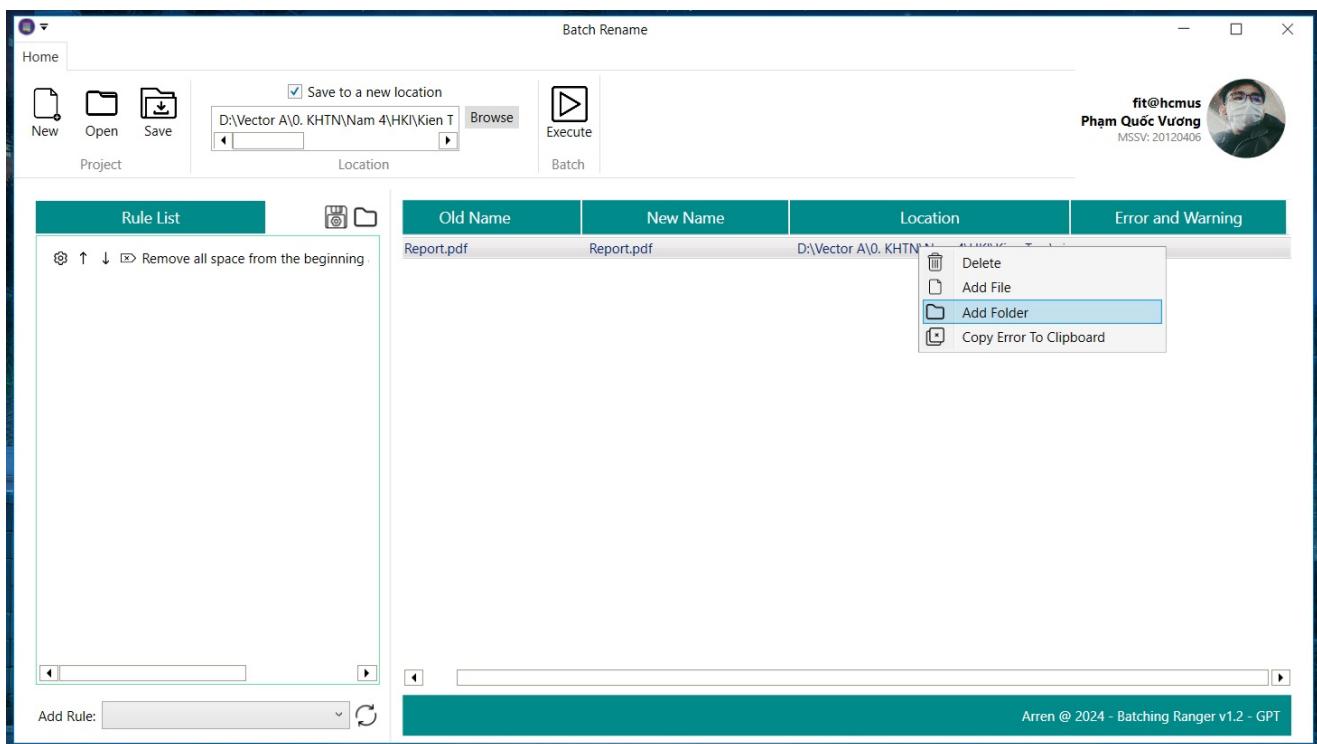
- Giao diện chỉnh sửa các tham số của các luật:



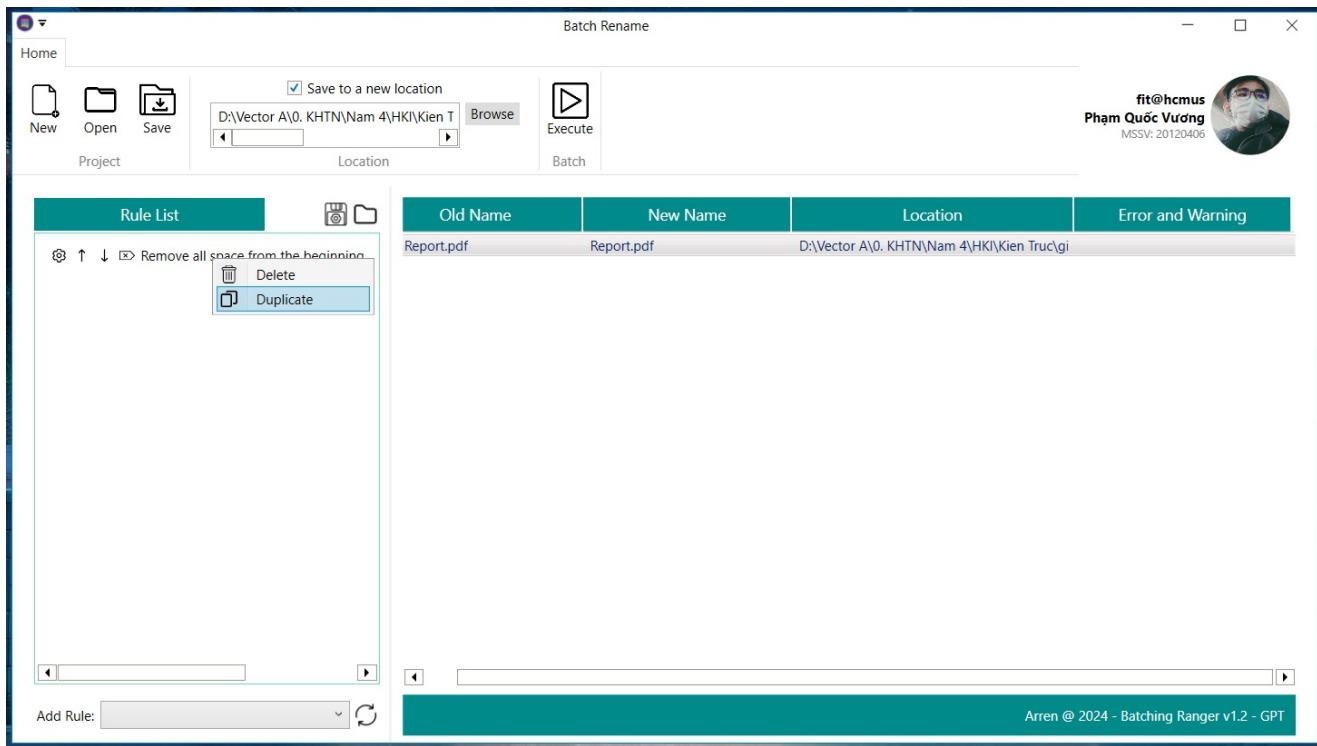
- Thông báo lỗi khi tham số nhập vào không đúng định dạng (data validation):



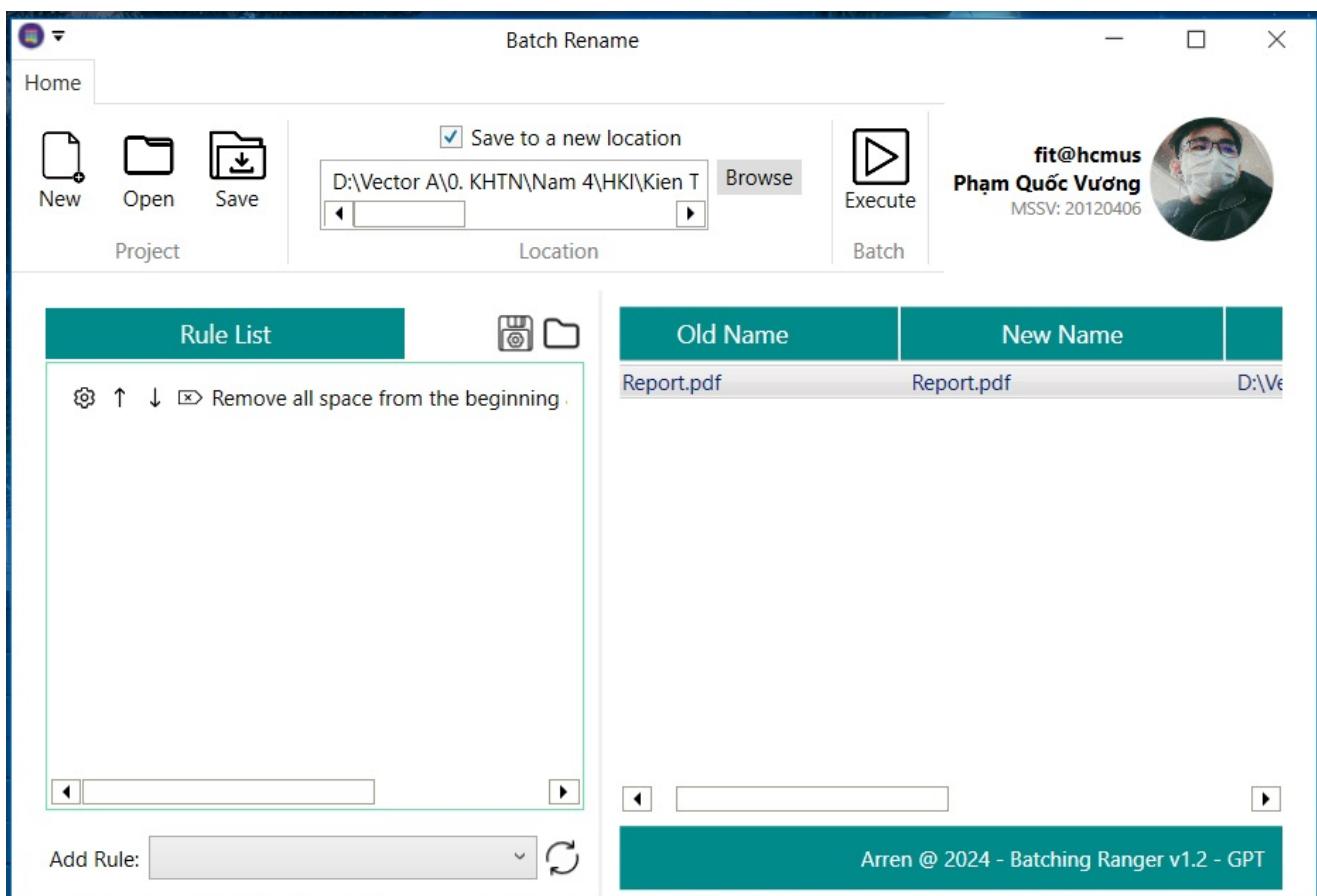
- Context menu khi nhấp chuột phải ở một file/folder item bất kỳ:



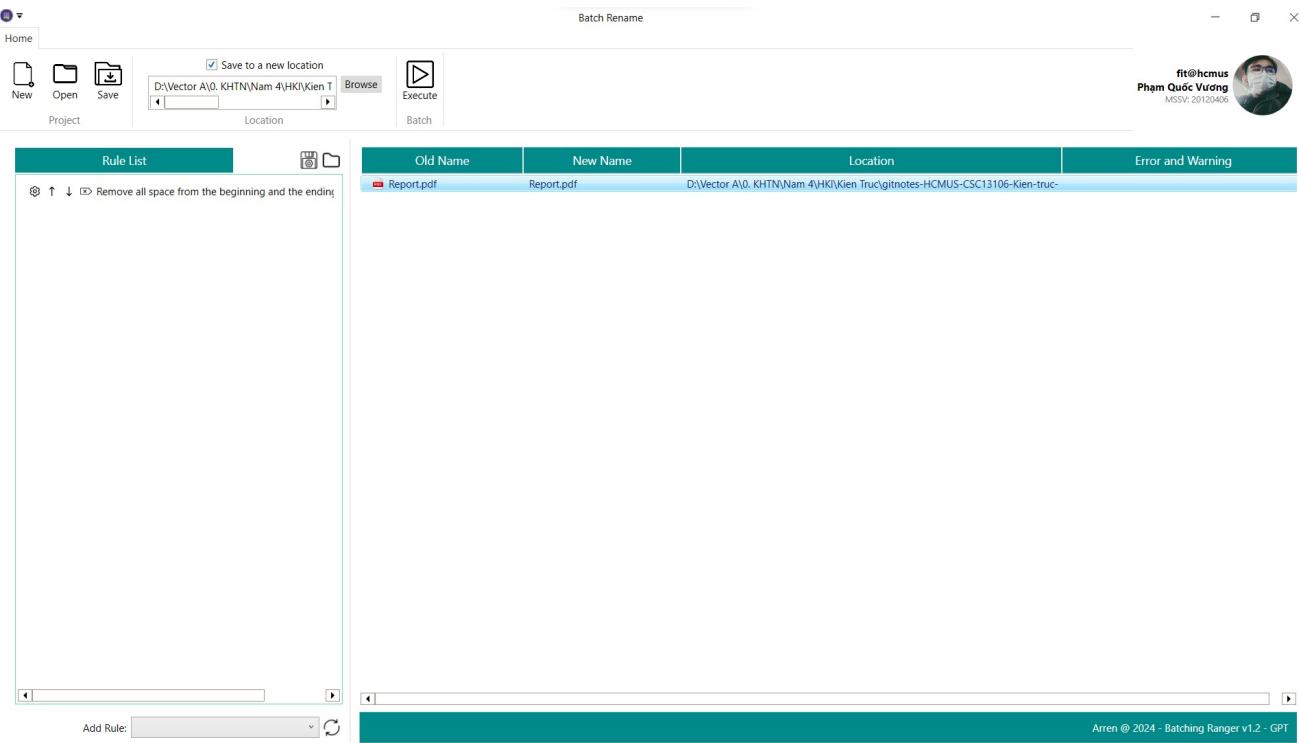
- Context menu khi nhấp chuột phải ở một luật đổi tên bất kỳ:



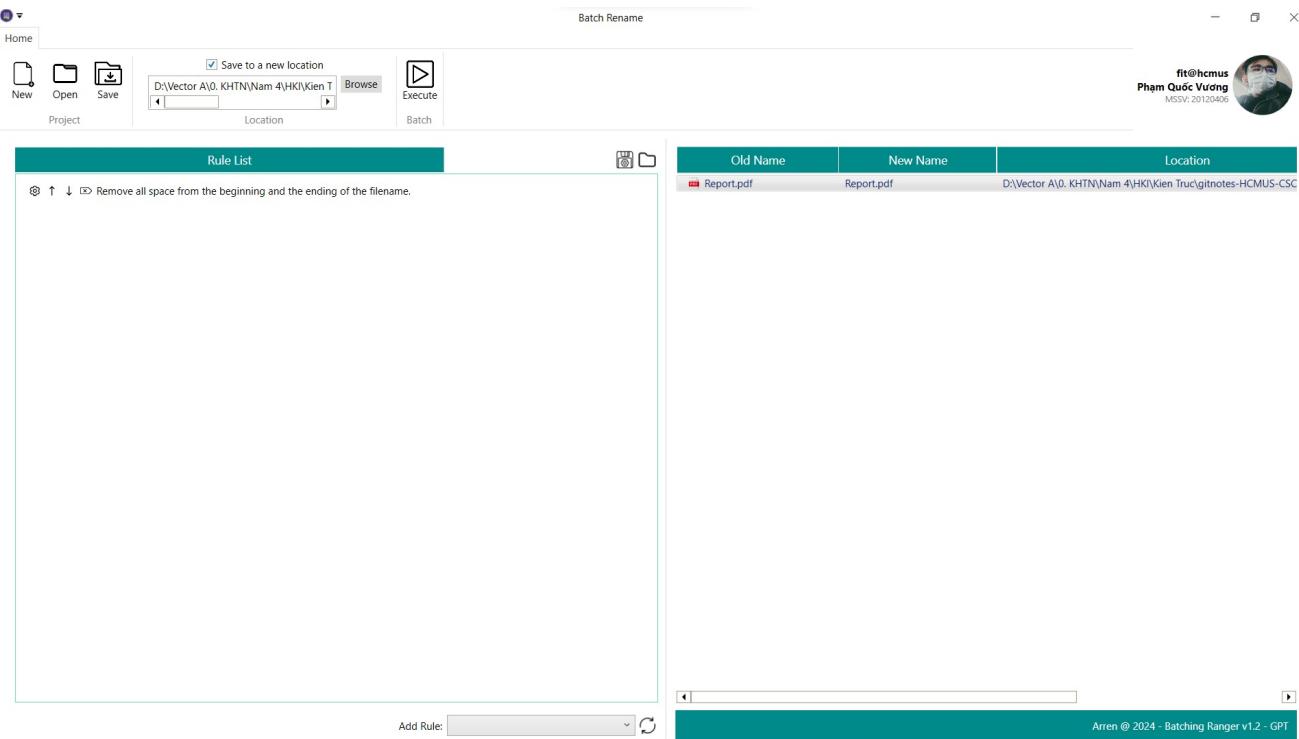
- Giao diện khi co lại mức nhỏ nhất:



- Giao diện khi ở chế độ full screen:



- Các khu vực màn hình có thể điều chỉnh được:



#### 4 Link video demo:

Xem video demo [tại đây](#).

#### 5 Kiến trúc và kỹ thuật nổi bật

- Kiến trúc tổng quát gồm:

- Một ứng dụng chính

- Một thư mục gồm các file DLL chứa các luật đổi tên để ứng dụng chính đọc và thêm vào chương trình. Tên pattern được áp dụng cho kiểu xử lý này: **Dependency Injection** (một kiểu **Inversion of control**).

The screenshot shows the Visual Studio IDE. In the center, the code editor displays `DLLUtils.cs` with the following content:

```

7     {
8         public static Tuple<bool, string, List<InterfaceType>>
9             getAllImplementableClassInstanceFromDLLs<InterfaceType>(
10            string dllFolder
11        ) {
12            bool success = true;
13            string message = "";
14
15            List<InterfaceType> instances = new List<InterfaceType>();
16
17            try {
18                var folderInfo = new DirectoryInfo(dllFolder);
19                var dllFiles = folderInfo.GetFiles("*.dll");
20
21                foreach (var file in dllFiles) {
22                    var assembly = Assembly.LoadFrom(file.FullName);
23                    var types = assembly.GetTypes();
24
25                    foreach (var type in types) {
26                        if (type.IsClass && typeof(InterfaceType).IsAssignableFrom(type))
27                        {
28                            InterfaceType rule = (InterfaceType)Activator
29                                .CreateInstance(type);
30
31                            instances.Add(rule);
32
33                        }
34
35                    }
36
37                }
38
39                success = true;
40            }
41            catch (Exception ex) {
42                {
43                    success = false;
44                    message = string.Concat(
45                        ex.Message,
46                        ": ",
47                        ex.StackTrace
48                    );
49
50                }
51
52                Tuple<bool, string, List<InterfaceType>> result
53                = new Tuple<bool, string, List<InterfaceType>>(
54                    success,
55                    message,
56                    instances
57                );
58
59            }
60        }
61    }

```

To the right of the code editor is the Solution Explorer window, which lists the project structure:

- BatchRename\_MainUI**
  - Dependencies
  - library
  - resource
    - app.manifest
    - AssemblyInfo.cs
    - FodyWeavers.xml
    - LocationAndFileNameToFileIconConve
  - MainWindow.xaml
  - MainWindow.xaml.cs
  - RuleSettingUI.xaml
- BatchRename\_UtilClasses**
  - Dependencies
  - BindableRuleParameter.cs
  - DLLUtils.cs
  - Factory.cs
  - FodyWeavers.xml
  - JSONUtils.cs
  - NameUtils.cs
  - PathUtils.cs
  - PresetUtils.cs
  - ProjectUtils.cs
  - Rule\_AddCounterToTheEnd
  - Rule\_AddingPrefix
  - Rule\_AddingSuffix
  - Rule\_ChangeExtension
  - Rule\_GPT

- Mẫu **Factory**, **Singleton** và **Prototype** kết hợp với ý tưởng **Template Class/Generic Class** hỗ trợ trong việc khởi tạo các luật, cũng như giúp việc sử dụng Factory về sau dễ dàng nếu cần áp dụng lên kiểu dữ liệu mới.

The screenshot shows the Visual Studio IDE. In the center, the code editor displays `Factory.cs` with the following content:

```

10    10 references
11    11
12    12
13    13
14    14
15    15
16    16
17    17
18    18
19    19
20    20
21    21
22    22
23    23
24    24
25    25
26    26
27    27
28    28
29    29
30    30
31    31
32    32
33    33
34    34
35    35
36    36
37    37
38    38
39    39
40    40
41    41
42    42
43    43
44    44
45    45
46    46
47    47
48    48
49    49
50    50
51    51
52    52
53    53
54    54
55    55
56    56
57    57
58    58
59    59
60    60
61    61
62    62
63    63
64    64
65    65
66    66
67    67
68    68
69    69
70    70
71    71
72    72
73    73
74    74
75    75
76    76
77    77
78    78
79    79
80    80
81    81
82    82
83    83
84    84
85    85
86    86
87    87
88    88
89    89
90    90
91    91
92    92
93    93
94    94
95    95
96    96
97    97
98    98
99    99
100   100
101   101
102   102
103   103
104   104
105   105
106   106
107   107
108   108
109   109
110   110
111   111
112   112
113   113
114   114
115   115
116   116
117   117
118   118
119   119
120   120
121   121
122   122
123   123
124   124
125   125
126   126
127   127
128   128
129   129
130   130
131   131
132   132
133   133
134   134
135   135
136   136
137   137
138   138
139   139
140   140
141   141
142   142
143   143
144   144
145   145
146   146
147   147
148   148
149   149
150   150
151   151
152   152
153   153
154   154
155   155
156   156
157   157
158   158
159   159
160   160
161   161
162   162
163   163
164   164
165   165
166   166
167   167
168   168
169   169
170   170
171   171
172   172
173   173
174   174
175   175
176   176
177   177
178   178
179   179
180   180
181   181
182   182
183   183
184   184
185   185
186   186
187   187
188   188
189   189
190   190
191   191
192   192
193   193
194   194
195   195
196   196
197   197
198   198
199   199
200   200
201   201
202   202
203   203
204   204
205   205
206   206
207   207
208   208
209   209
210   210
211   211
212   212
213   213
214   214
215   215
216   216
217   217
218   218
219   219
220   220
221   221
222   222
223   223
224   224
225   225
226   226
227   227
228   228
229   229
230   230
231   231
232   232
233   233
234   234
235   235
236   236
237   237
238   238
239   239
240   240
241   241
242   242
243   243
244   244
245   245
246   246
247   247
248   248
249   249
250   250
251   251
252   252
253   253
254   254
255   255
256   256
257   257
258   258
259   259
260   260
261   261
262   262
263   263
264   264
265   265
266   266
267   267
268   268
269   269
270   270
271   271
272   272
273   273
274   274
275   275
276   276
277   277
278   278
279   279
280   280
281   281
282   282
283   283
284   284
285   285
286   286
287   287
288   288
289   289
290   290
291   291
292   292
293   293
294   294
295   295
296   296
297   297
298   298
299   299
300   300
301   301
302   302
303   303
304   304
305   305
306   306
307   307
308   308
309   309
310   310
311   311
312   312
313   313
314   314
315   315
316   316
317   317
318   318
319   319
320   320
321   321
322   322
323   323
324   324
325   325
326   326
327   327
328   328
329   329
330   330
331   331
332   332
333   333
334   334
335   335
336   336
337   337
338   338
339   339
340   340
341   341
342   342
343   343
344   344
345   345
346   346
347   347
348   348
349   349
350   350
351   351
352   352
353   353
354   354
355   355
356   356
357   357
358   358
359   359
360   360
361   361
362   362
363   363
364   364
365   365
366   366
367   367
368   368
369   369
370   370
371   371
372   372
373   373
374   374
375   375
376   376
377   377
378   378
379   379
380   380
381   381
382   382
383   383
384   384
385   385
386   386
387   387
388   388
389   389
390   390
391   391
392   392
393   393
394   394
395   395
396   396
397   397
398   398
399   399
400   400
401   401
402   402
403   403
404   404
405   405
406   406
407   407
408   408
409   409
410   410
411   411
412   412
413   413
414   414
415   415
416   416
417   417
418   418
419   419
420   420
421   421
422   422
423   423
424   424
425   425
426   426
427   427
428   428
429   429
430   430
431   431
432   432
433   433
434   434
435   435
436   436
437   437
438   438
439   439
440   440
441   441
442   442
443   443
444   444
445   445
446   446
447   447
448   448
449   449
450   450
451   451
452   452
453   453
454   454
455   455
456   456
457   457
458   458
459   459
460   460
461   461
462   462
463   463
464   464
465   465
466   466
467   467
468   468
469   469
470   470
471   471
472   472
473   473
474   474
475   475
476   476
477   477
478   478
479   479
480   480
481   481
482   482
483   483
484   484
485   485
486   486
487   487
488   488
489   489
490   490
491   491
492   492
493   493
494   494
495   495
496   496
497   497
498   498
499   499
500   500
501   501
502   502
503   503
504   504
505   505
506   506
507   507
508   508
509   509
510   510
511   511
512   512
513   513
514   514
515   515
516   516
517   517
518   518
519   519
520   520
521   521
522   522
523   523
524   524
525   525
526   526
527   527
528   528
529   529
530   530
531   531
532   532
533   533
534   534
535   535
536   536
537   537
538   538
539   539
540   540
541   541
542   542
543   543
544   544
545   545
546   546
547   547
548   548
549   549
550   550
551   551
552   552
553   553
554   554
555   555
556   556
557   557
558   558
559   559
560   560
561   561
562   562
563   563
564   564
565   565
566   566
567   567
568   568
569   569
570   570
571   571
572   572
573   573
574   574
575   575
576   576
577   577
578   578
579   579
580   580
581   581
582   582
583   583
584   584
585   585
586   586
587   587
588   588
589   589
590   590
591   591
592   592
593   593
594   594
595   595
596   596
597   597
598   598
599   599
600   600
601   601
602   602
603   603
604   604
605   605
606   606
607   607
608   608
609   609
610   610
611   611
612   612
613   613
614   614
615   615
616   616
617   617
618   618
619   619
620   620
621   621
622   622
623   623
624   624
625   625
626   626
627   627
628   628
629   629
630   630
631   631
632   632
633   633
634   634
635   635
636   636
637   637
638   638
639   639
640   640
641   641
642   642
643   643
644   644
645   645
646   646
647   647
648   648
649   649
650   650
651   651
652   652
653   653
654   654
655   655
656   656
657   657
658   658
659   659
660   660
661   661
662   662
663   663
664   664
665   665
666   666
667   667
668   668
669   669
670   670
671   671
672   672
673   673
674   674
675   675
676   676
677   677
678   678
679   679
680   680
681   681
682   682
683   683
684   684
685   685
686   686
687   687
688   688
689   689
690   690
691   691
692   692
693   693
694   694
695   695
696   696
697   697
698   698
699   699
700   700
701   701
702   702
703   703
704   704
705   705
706   706
707   707
708   708
709   709
710   710
711   711
712   712
713   713
714   714
715   715
716   716
717   717
718   718
719   719
720   720
721   721
722   722
723   723
724   724
725   725
726   726
727   727
728   728
729   729
730   730
731   731
732   732
733   733
734   734
735   735
736   736
737   737
738   738
739   739
740   740
741   741
742   742
743   743
744   744
745   745
746   746
747   747
748   748
749   749
750   750
751   751
752   752
753   753
754   754
755   755
756   756
757   757
758   758
759   759
760   760
761   761
762   762
763   763
764   764
765   765
766   766
767   767
768   768
769   769
770   770
771   771
772   772
773   773
774   774
775   775
776   776
777   777
778   778
779   779
780   780
781   781
782   782
783   783
784   784
785   785
786   786
787   787
788   788
789   789
790   790
791   791
792   792
793   793
794   794
795   795
796   796
797   797
798   798
799   799
800   800
801   801
802   802
803   803
804   804
805   805
806   806
807   807
808   808
809   809
810   810
811   811
812   812
813   813
814   814
815   815
816   816
817   817
818   818
819   819
820   820
821   821
822   822
823   823
824   824
825   825
826   826
827   827
828   828
829   829
830   830
831   831
832   832
833   833
834   834
835   835
836   836
837   837
838   838
839   839
840   840
841   841
842   842
843   843
844   844
845   845
846   846
847   847
848   848
849   849
850   850
851   851
852   852
853   853
854   854
855   855
856   856
857   857
858   858
859   859
860   860
861   861
862   862
863   863
864   864
865   865
866   866
867   867
868   868
869   869
870   870
871   871
872   872
873   873
874   874
875   875
876   876
877   877
878   878
879   879
880   880
881   881
882   882
883   883
884   884
885   885
886   886
887   887
888   888
889   889
890   890
891   891
892   892
893   893
894   894
895   895
896   896
897   897
898   898
899   899
900   900
901   901
902   902
903   903
904   904
905   905
906   906
907   907
908   908
909   909
910   910
911   911
912   912
913   913
914   914
915   915
916   916
917   917
918   918
919   919
920   920
921   921
922   922
923   923
924   924
925   925
926   926
927   927
928   928
929   929
930   930
931   931
932   932
933   933
934   934
935   935
936   936
937   937
938   938
939   939
940   940
941   941
942   942
943   943
944   944
945   945
946   946
947   947
948   948
949   949
950   950
951   951
952   952
953   953
954   954
955   955
956   956
957   957
958   958
959   959
960   960
961   961
962   962
963   963
964   964
965   965
966   966
967   967
968   968
969   969
970   970
971   971
972   972
973   973
974   974
975   975
976   976
977   977
978   978
979   979
980   980
981   981
982   982
983   983
984   984
985   985
986   986
987   987
988   988
989   989
990   990
991   991
992   992
993   993
994   994
995   995
996   996
997   997
998   998
999   999
1000  1000
1001  1001
1002  1002
1003  1003
1004  1004
1005  1005
1006  1006
1007  1007
1008  1008
1009  1009
1010  1010
1011  1011
1012  1012
1013  1013
1014  1014
1015  1015
1016  1016
1017  1017
1018  1018
1019  1019
1020  1020
1021  1021
1022  1022
1023  1023
1024  1024
1025  1025
1026  1026
1027  1027
1028  1028
1029  1029
1030  1030
1031  1031
1032  1032
1033  1033
1034  1034
1035  1035
1036  1036
1037  1037
1038  1038
1039  1039
1040  1040
1041  1041
1042  1042
1043  1043
1044  1044
1045  1045
1046  1046
1047  1047
1048  1048
1049  1049
1050  1050
1051  1051
1052  1052
1053  1053
1054  1054
1055  1055
1056  1056
1057  1057
1058  1058
1059  1059
1060  1060
1061  1061
1062  1062
1063  1063
1064  1064
1065  1065
1066  1066
1067  1067
1068  1068
1069  1069
1070  1070
1071  1071
1072  1072
1073  1073
1074  1074
1075  1075
1076  1076
1077  1077
1078  1078
1079  1079
1080  1080
1081  1081
1082  1082
1083  1083
1084  1084
1085  1085
1086  1086
1087  1087
1088  1088
1089  1089
1090  1090
1091  1091
1092  1092
1093  1093
1094  1094
1095  1095
1096  1096
1097  1097
1098  1098
1099  1099
1100  1100
1101  1101
1102  1102
1103  1103
1104  1104
1105  1105
1106  1106
1107  1107
1108  1108
1109  1109
1110  1110
1111  1111
1112  1112
1113  1113
1114  1114
1115  1115
1116  1116
1117  1117
1118  1118
1119  1119
1120  1120
1121  1121
1122  1122
1123  1123
1124  1124
1125  1125
1126  1126
11
```

**Factory.cs**

```

public Tuple<bool, string, InterfaceType?> Produce(string ClassName)
{
    bool success = true;
    string message = "";
    InterfaceType? result;

    try
    {
        if (_prototypes.ContainsKey(ClassName))
        {
            InterfaceType prototype = _prototypes[ClassName];
            result = (InterfaceType)prototype.Clone();
            success = true;
        }
        else
        {
            success = false;
            message = $"Internal error: the rule with ClassName={ClassName} might not be registered!";
            result = default(InterfaceType);
        }
    }
    catch(Exception e)
    {
        success = false;
        message = string.Concat(e.Message, e.StackTrace);
        result = default(InterfaceType);
    }

    return new Tuple<bool, string, InterfaceType?>(
        success,
        message,
        result
    );
}

```

**Solution Explorer**

- App.xaml
- AssemblyInfo.cs
- FodyWeavers.xml
- MainWindow.xaml
- MainWindow.xaml.cs
- RuleSettingU.xaml
- BatchRename\_UtilClasses
- Dependencies
- BindableRuleParameter.cs
- DLLUtils.cs
- Factory.cs
- FodyWeavers.xml
- JSONUtils.cs
- NameUtils.cs
- PathUtils.cs
- PresetUtils.cs
- ProjectUtils.cs
- Rule\_AddCounterToTheEnd
- Rule\_AddingPrefix
- Rule\_AddingSuffix
- Rule\_ChangeExtension
- Rule\_GPT
- Dependencies
- app.manifest
- GPTRule.cs

- Sử dụng kỹ thuật **serialization** và **deserialization** (tuần tự hóa và giải tuần tự hóa) để hỗ trợ việc lưu và đọc dữ liệu một instance của class bằng kiểu dữ liệu JSON. Ngoài ra, các thuộc tính không được phép thay đổi do hệ thống quy định được gán metadata là [JsonIgnore] để tránh tình trạng người dùng sửa dữ liệu trong file làm quá trình deserialize gây thay đổi không mong muốn.

**JSONUtils.cs**

```

public static Tuple<bool, string, T?> jsonFileReader(string filepath)
{
    bool success = true;
    string message = "";

    var jsonString = File.ReadAllText(filepath);

    T? readResult = default(T);

    try
    {
        readResult = JsonSerializer.Deserialize<T>(jsonString);

        if (readResult == null)
        {
            success = false;
            message = string.Concat(
                "Either null or empty ",
                typeof(T).Name,
                " data was read from ",
                filepath
            );
        }
        else
        {
            success = true;
        }
    }
    catch (Exception ex)
    {
        success = false;
        message = string.Concat(
            ex.Message,
            ":",
            ex.StackTrace
        );
    }

    return new Tuple<bool, string, T?>(
        success,
        message,
        readResult
    );
}

```

**Solution Explorer**

- Project.cs
- BatchRename\_MainUI
  - Dependencies
  - library
  - resource
  - app.manifest
  - App.xaml
  - AssemblyInfo.cs
  - FodyWeavers.xml
  - LocationAndFileNameToFileIconConv
  - MainWindow.xaml
  - MainWindow.xaml.cs
  - RuleSettingU.xaml
- BatchRename\_UtilClasses
- Dependencies
- BindableRuleParameter.cs
- DLLUtils.cs
- Factory.cs
- FodyWeavers.xml
- JSONUtils.cs
- NameUtils.cs
- PathUtils.cs
- PresetUtils.cs
- ProjectUtils.cs
- Rule\_AddCounterToTheEnd
- Rule\_AddingPrefix
- Rule\_AddingSuffix
- Rule\_ChangeExtension
- Rule\_GPT
- Dependencies
- app.manifest
- GPTRule.cs
- GPTUtils.cs
- JSONUtils.cs

The screenshot shows the Visual Studio IDE interface. The code editor displays the `JSONUtils.cs` file, which contains C# code for serializing objects to JSON and writing them to a file. The Solution Explorer on the right shows the project structure for the `BatchRename` solution, including files like `MainWindow.xaml.cs`, `RuleSettingUI.xaml`, and various utility classes.

```

    public static Tuple<bool, string> jsonFileWriter(string filepath, T data)
    {
        bool success = true;
        string message = "";

        if (data == null)
        {
            success = false;
            message = string.Concat(
                "Either null or empty ",
                typeof(T).ToString(),
                " data"
            );
        }
        else
        {
            try
            {
                var jsonStringOptions = new JsonSerializerOptions
                {
                    WriteIndented = true
                };

                string jsonString = JsonSerializer.Serialize<T>(data,
                    jsonStringOptions
                );

                StreamWriter stream = new StreamWriter(filepath);

                stream.Write(jsonString);
                stream.Close();

                success = true;
            }
            catch (Exception ex)
            {
                success = false;
                message = string.Concat(
                    ex.Message,
                    ";",
                    ex.StackTrace
                );
            }
        }

        return new Tuple<bool, string>(success, message);
    }
}

```

The screenshot shows the Visual Studio IDE interface. The code editor displays the `IRule.cs` file, which defines an interface for rules. It includes properties for rule name and parameters, and methods for validating parameters and renaming files. The Solution Explorer on the right shows the project structure for the `BatchRename` solution, including files like `MainWindow.xaml.cs`, `RuleSettingUI.xaml`, and various utility classes.

```

    public interface IRule : INotifyPropertyChanged, ICloneable
    {
        public string RuleName { get; }

        public Dictionary<string, string> RuleParameters { get; set; }

        [JsonIgnore]
        public bool ShouldAnyDialogBeOpened { get; }

        [JsonIgnore]
        public string TextToDisplay { get; }

        public Tuple<bool, string> ValidateParameters(string paramKey, string paramValue);

        public Tuple<bool, string, string> Rename(string oldName, bool isFolder);

        [JsonIgnore]
        public bool HaveGlobalAffect { get; }

        public void ResetGlobalParameter();
    }
}

```

- Bổ sung tính năng **gọi API từ OpenAI** để xây dựng luật sửa lỗi chính tả và xóa ký tự thừa/ký tự không hợp lệ trong tên file. Với việc tích hợp luật đổi tên mới, sinh viên đã thực hiện rất trơn tru. Điều này chứng minh kiến trúc cũ đã được build đủ tốt để có thể mở rộng ứng dụng dễ dàng.

GPTUtils.cs

```

1 reference
public class GPTUtils
{
    private static List<string> apiKeyList = new List<string>
    {
        "sk-MkqE7NJZ7Fc", "Imh6",
        "sk-CHSI5bXp0z\Jfmp", "SDqV"
    };

    private const string apiUrl = "https://api.openai.com/v1/chat/completions";
    private const string openaiModel = "gpt-3.5-turbo";

    public static async Task<Tuple<bool, string, OpenAIAPIDTO>> Prompt(
        string userPrompt,
        double temperature = 0.7
    ) {
        bool success = true;
        string message = "";
        OpenAIAPIDTO? requestResult = new OpenAIAPIDTO();

        Random random = new Random();
        int index = random.Next(0, apiKeyList.Count); // Rate limiting issue
        string selectedKey = apiKeyList[index];

        using (var client = new HttpClient())
        {
            // Request headers
            client.DefaultRequestHeaders.Add("Authorization", $"Bearer {selectedKey}");

            // Request body content
            var body = string.Concat(
                $"\"{\"model\": \"{openaiModel}\",",
                $"\"messages\": [{\"role\": \"user\", \"content\": \"{userPrompt}\", \"temperature\": {temperature}}]}"
            );
        }

        string mediaType = "application/json";

        var content = new StringContent(body, Encoding.UTF8, mediaType);
    }
}

```

Solution Explorer

- FodyWeavers.xml
- C# LocationAndFileNameToFileIconConversion
- MainWindow.xaml
- RuleSettingUI.xaml
- BatchRename\_UtilClasses
  - Dependencies
  - BindableRuleParameter.cs
  - DLLUtils.cs
  - Factory.cs
    - FodyWeavers.xml
  - JSONUtils.cs
  - NameUtils.cs
  - PathUtils.cs
  - PresetUtils.cs
  - ProjectUtils.cs
- Rule\_GPT
  - Dependencies
  - app.manifest
  - GPTRule.cs
  - GPTUtils.cs
  - JSONUtils.cs
  - OpenAIAPIDTO.cs
- Rule\_MakeAllLowercasedAndNoWhiteSpace
  - Dependencies
  - app.manifest
  - MakeAllLowercasedAndNoWhiteSpace
- Rule\_PascalCase
- Rule\_RemoveBeginningAndEndingSpace
- Rule\_ReplaceCharacters
- TestingConsole
- Setup

GPTUtils.cs

```

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83

```

```

string mediaType = "application/json";

var content = new StringContent(body, Encoding.UTF8, mediaType);

try
{
    // Call API
    HttpResponseMessage response = client.PostAsync(apiUrl, content).Result;
    //var response = await client.PostAsync(apiUrl, content);

    // Check the status code of the response
    if (response.IsSuccessStatusCode)
    {
        string responseContent = await response.Content.ReadAsStringAsync();

        responseContent = responseContent.Replace("\\"object\": \"", "\\"object_info\": \"");

        (success, message, requestResult) = JSONUtils<OpenAIAPIDTO>.Deserialize(responseContent);
    }
    else
    {
        success = false;
        message = $"Error in Prompt function: {response.StatusCode} - {response.ReasonPhrase}";
        requestResult = null;
    }
}
catch (Exception ex)
{
    success = false;
    message = string.Concat(
        "You might be disconnected from the internet. Error: ",
        ex.Message
    );
    requestResult = null;
}

return new Tuple<bool, string, OpenAIAPIDTO>(
    success,
    message,
    requestResult!
);

```

Solution Explorer

- FodyWeavers.xml
- C# LocationAndFileNameToFileIconConversion
- MainWindow.xaml
- RuleSettingUI.xaml
- BatchRename\_UtilClasses
  - Dependencies
  - BindableRuleParameter.cs
  - DLLUtils.cs
  - Factory.cs
    - FodyWeavers.xml
  - JSONUtils.cs
  - NameUtils.cs
  - PathUtils.cs
  - PresetUtils.cs
  - ProjectUtils.cs
- Rule\_GPT
  - Dependencies
  - app.manifest
  - GPTRule.cs
  - GPTUtils.cs
  - JSONUtils.cs
  - OpenAIAPIDTO.cs
- Rule\_MakeAllLowercasedAndNoWhiteSpace
  - Dependencies
  - app.manifest
  - MakeAllLowercasedAndNoWhiteSpace
- Rule\_PascalCase
- Rule\_RemoveBeginningAndEndingSpace
- Rule\_ReplaceCharacters
- TestingConsole
- Setup

```

private async Task<Tuple<bool, string, string>> AsyncRename(string oldName, bool isFolder)
{
    bool success = true;
    string message = "";
    string result = oldName;

    string name = "";
    string extension = "";

    if (!string.IsNullOrEmpty(oldName))
    {
        if (isFolder)
        {
            name = oldName;
            extension = "";
        }
        else
        {
            var extensionSplitter = oldName.Split('.');
            extension = extensionSplitter.Last();
            name = string.Join(".", extensionSplitter.SkipLast(1));
        }

        name = name.Replace(oldValue: "\\", newValue: "");
    }

    var prompt = string.Concat(
        $"My file name is '{name}'",
        "(the file extension is hidden).",
        "Let's detect errors in this (like",
        "grammar errors,",
        "redundant letters,",
        "inappropriate characters for file name like",
        "colons, ?, slashes, semicolons, commas, +, =, <, >, quotes, double quotes, |, [, ], etc',",
        "then give me a new file name with these errors fixed",
        "(the new name should be based on the old name)",
        "(place that new file name between two square brackets",
        "like this: 'The new file name is: [place_the_new_file_name_here] ',",
        "so that I can use code to parse it)",
        "(of course, with no file extension)"
    );

    OpenAIAPIDTO openAIAPIDTO = new OpenAIAPIDTO();

```

- Sử dụng **Regular Expression** để tìm chuỗi mong muốn, validate dữ liệu, v.v.

```

"like this: 'The new file name is: [place_the_new_file_name_here] ',",
"so that I can use code to parse it",
"(of course, with no file extension"
);

OpenAIAPIDTO openAIAPIDTO = new OpenAIAPIDTO();
(success, message, openAIAPIDTO) = await GPTUtils.Prompt(prompt);

if (success)
{
    string responseMessage = openAIAPIDTO.choices[0].message.content;

    // [...] or "... or '...
    string pattern = @"([([^\?\\;,\+\=\>\^\|;\\[\]\^\]+])|(\^\([^\?\\;,\+\=\>\^\|;\\[\]\^\]+)\)|(\^\([^\?\\;,\+\=\>\^\|;\\[\]\^\]+\^\|;\\[\]\^\)+\^\|";
    Regex regex = new Regex(pattern);

    Match match = regex.Match(responseMessage);

    if (match.Success)
    {
        success = true;
        name = match.Value;
        name = name.Substring(startIndex: 1, length: name.Length - 2);
    }
    else
    {
        success = false;
        message = $"The response from GPT might be in a wrong format: {responseMessage}";
    }
}
else
{
    // Do nothing
}

```

```

public Tuple<bool, string> ValidateParameters(string paramKey, string paramValue)
{
    bool isValid = true;
    string message = "";

    if (paramKey == "Start" || paramKey == "Step")
    {
        Regex intRegex = new Regex("^-?\\d+$"); // integer
        isValid = intRegex.IsMatch(paramValue);

        if (!isValid)
        {
            message = "This field should be an integer";
        }
        else
        {
            // Do nothing
        }
    }
    else if (paramKey == "The number of digits")
    {
        Regex uintRegex = new Regex("^\\d+$"); // unsigned integer
        isValid = uintRegex.IsMatch(paramValue);

        if (!isValid)
        {
            message = "This field should be an unsigned integer";
        }
        else
        {
            // Do nothing
        }
    }

    return new Tuple<bool, string>
    (
        isValid,
        message
    );
}

```

## 6 Các chức năng đã hoàn thiện

### 6.1 Chức năng chính

- Load luật đổi tên từ các file DLL (có thể refresh lại danh sách luật khi chương trình đang chạy)
- Thêm file hoặc folder cần đổi tên vào chương trình (có thể thêm hàng loạt file hoặc hàng loạt folder cùng lúc)
- Tạo danh sách các luật đổi để áp dụng lên danh sách file và folder để thêm vào:
  - Mỗi luật được chọn và thêm vào từ dropdown menu
  - Các luật có parameter đầu vào sẽ có màn hình để thay đổi các parameter đó (các parameter khi bị nhập giá trị không hợp lệ sẽ hiện thông báo lỗi)
  - Thứ tự các luật có thể hoán đổi với nhau
  - Có thể xóa luật mong muốn khỏi danh sách
  - Có thể nhân bản luật mong muốn trong danh sách
- Lần lượt từng file sẽ được áp dụng danh sách các luật đổi tên theo thứ tự được thêm vào (file được thêm vào trước sẽ được áp dụng luật đổi tên trước)
- Lần lượt từng folder sẽ được áp dụng danh sách các luật đổi tên theo thứ tự được thêm vào (folder được thêm vào trước sẽ được áp dụng luật đổi tên trước)
- Danh sách các luật có thể được lưu thành file preset để tái sử dụng
- Có thể lưu trạng thái công việc hiện tại thành file project để mở lên làm việc tiếp vào lúc khác
- Cho phép người dùng preview kết quả đổi tên trước khi quyết định thực thi hành động đổi tên

### 6.2 Các luật đổi tên

- Rule\_AddCounterToTheEnd.dll:** Thêm số đếm vào đuôi file:
  - Với start value, steps, number of digits được chỉ định trước
  - Hỗ trợ thêm padding, ví dụ: 01, 02, 03, ..., 10, ..., 99

2. **Rule\_RemoveBeginningAndEndingSpace.dll:** Xóa khoảng trắng ở đầu tên file và cuối tên file
3. **Rule\_ReplaceCharacters.dll:** Đổi ký tự a trong tên file thành ký tự b được chỉ định
4. **Rule\_AddingPrefix.dll:** Thêm tiền tố (prefix) vào đầu mỗi file
5. **Rule\_AddingSuffix.dll:** Thêm hậu tố (suffix) vào cuối mỗi file
6. **Rule\_ChangeExtension.dll:** Đổi file extension theo mong muốn (chỉ đổi tên extension, không convert)
7. **Rule\_GPT.dll:** sửa lỗi ngữ pháp, lỗi ký tự dư thừa và ký tự không hợp lệ trong tên file theo gợi ý của ChatGPT (sử dụng API của OpenAI để lấy gợi ý)
8. **Rule\_MakeAllLowercasedAndNoWhiteSpace.dll:** Lowercase tất cả ký tự trong tên file và xóa hết khoảng trắng
9. **Rule\_PascalCase.dll:** Chuyển tên file thành dạng PascalCase

### 6.3 Các chức năng/kỹ thuật khác

1. Xử lý trùng tên: sau khi áp dụng luật, nếu các file/folder ở cùng folder mẹ có tên trùng nhau → đánh số (0), (1), (2), ... ở cuối các file/folder trùng nhau
2. Dùng regular expressions → sử dụng trong code để kiểm tra một số loại chuỗi có cấu trúc nhất định (khác với việc sử dụng vào việc cho người dùng nhập Regex và dựa vào đó để đổi tên)
3. Cho phép người dùng chọn vị trí mới để lưu file/folder hoặc lưu ở vị trí ban đầu của file/folder
4. Nếu người dùng chọn lưu file được đổi tên vào thư mục mới, trước khi đổi tên, nếu thư mục đích có file trùng tên với file sắp được lưu, chương trình sẽ di chuyển các file cũ đến thư mục chứa file trùng mặc định của chương trình nằm cùng cấp với file .exe. Bên trong thư mục chứa file trùng là các thư mục có tên là ngày giờ, bên trong mỗi thư mục sẽ là nội dung trùng được chuyển đến.
5. Lấy icon gốc của các file được chọn để hiển thị lên chương trình, tăng tính thẩm mỹ.
6. Giao diện chứa các vị trí cho phép kéo thả để co giãn các khu vực màn hình nhất định, có các thanh trượt để di chuyển khi cần.
7. Hệ thống context menu với các nút tiện ích hữu dụng (thêm/xóa file/folder khỏi danh sách, nút để copy lỗi của từng file cố định, xóa luật khỏi danh sách, nhân đôi/duplicate/clone một luật trong danh sách luật), nút reload để refresh lại danh sách dll được đọc từ thư mục **library** (ngang cấp với file .exe).
8. Các nút di chuyển lên/xuống, nút cài đặt luật, nút xóa luật nằm ngay trên danh sách luật giúp người dùng dễ tìm thấy và sử dụng.
9. Giao diện phóng khoáng, thân thiện.

## 7 Các lưu ý cần thực hiện để chạy chương trình

### 7.1 Lưu ý chung

- Tắt trình diệt virus và chức năng Controlled Folder Access (nếu chạy không thành công).

### 7.2 Chạy bằng bản no-install

Gọi **Release/no-install-version** là thư mục chứa file **BatchRename\_MainUI.exe**.

Đảm bảo thư mục **no-install-version** có cấu trúc cây thư mục như sau:

- no-install-version:

- library:
  - Rule\_AddCounterToTheEnd.dll
  - Rule\_AddingPrefix.dll
  - Rule\_AddingSuffix.dll
  - Rule\_ChangeExtension.dll
  - Rule\_GPT.dll
  - Rule\_MakeAllLowercasedAndNoWhiteSpace.dll
  - Rule\_PascalCase.dll
  - Rule\_RemoveBeginningAndEndingSpace.dll
  - Rule\_ReplaceCharacters.dll
- runtimes:
  - browser/lib/net6.0/System.Text.Encodings.Web.dll
- BatchRename\_MainUI.runtimeconfig.json
- BatchRename\_MainClasses.dll
- BatchRename\_MainUI.dll
- BatchRename\_MainUI.exe
- BatchRename\_UtilClasses.dll
- ControlzEx.dll
- Fluent.dll
- Microsoft.WindowsAPICodePack.dll
- Microsoft.WindowsAPICodePack.Shell.dll
- Microsoft.WindowsAPICodePack.ShellExtensions.dll
- Microsoft.Xaml.Behaviors.dll
- System.Text.Encodings.Web.dll
- System.Text.Json.dll

### 7.3 Chạy bằng file setup

Thư mục **Release/installer** có chứa các file cài đặt sau:

- setup.exe
- setup.msi

Có thể sử dụng một trong hai để cài đặt chương trình như các cài các chương trình phổ thông khác.

### 7.4 Chạy bằng project Visual Studio

- Mở file .sln.
- Chọn **BatchRename\_MainUI** làm startup project.
- Chạy bằng tổ hợp phím **Ctrl + F5**.

**Nếu chạy không thành công, thử check lại các bước sau:**

- Đảm bảo phiên bản phần mềm:
  - Microsoft Visual Studio Community 2022 (64-bit) - Current Version 17.8.2
  - Hoặc các bản có thể chạy project của bản vừa nêu.

- Trong project BatchRename\_MainUI:
  - Đảm bảo mọi tài nguyên ảnh trong thư mục **resource** đều được chọn Build Action là **Resource**.
  - Đảm bảo mọi tài nguyên .dll trong thư mục **library** đều được chọn Build Action là **None** và Copy to Output Directory là **Copy always**.
- Đảm bảo Project Dependencies được thiết lập như sau:
  - BatchRename\_MainClasses: không có phụ thuộc
  - BatchRename\_MainUI có phụ thuộc:
    - BatchRename\_MainClasses
    - BatchRename\_UtilClasses
  - BatchRename\_UtilClasses có phụ thuộc:
    - BatchRename\_MainClasses
  - Rule\_AddCounterToTheEnd, Rule\_AddingPrefix, Rule\_AddingSuffix, Rule\_ChangeExtension, Rule\_ReplaceCharacters có phụ thuộc:
    - BatchRename\_MainClasses
    - BatchRename\_UtilClasses
  - Rule\_GPT, Rule\_MakeAllLowercasedAndNoWhiteSpace, Rule\_PascalCase, Rule\_RemoveBeginningAndEndingSpace có phụ thuộc:
    - BatchRename\_MainClasses
- Cài các NuGet Packages:
  - PropertyChanged.Fody cho các project:
    - BatchRename\_MainClasses
    - BatchRename\_MainUI
    - BatchRename\_UtilClasses
    - Rule\_PascalCase
    - Rule\_RemoveBeginningAndEndingSpace
  - Fluent.Ribbon, Microsoft.WindowsAPICodePack-Shell, System.Text.Json cho project BatchRename\_MainUI
- Rebuild Solution và chạy lại bằng **Ctrl + F5**.
- Ngoài ra, nếu muốn dùng project Setup để tạo file **Setup.exe** hoặc **Setup.msi** cần:
  - Cài [Microsoft Visual Studio Installer Projects 2022](#).
  - Đảm bảo project có phụ thuộc đến BatchRename\_MainUI.