

```
1 // *****
2 // Assembly      : GeneralFunctionality
3 // Author        : G.H.M.H. Schmeits
4 // Created       : 01-05-2018
5 //
6 // Last Modified By : G.H.M.H. Schmeits
7 // Last Modified On : 05-15-2019
8 // *****
9 // <copyright file="Functions.cs" company="SCHMEITS SOFTWARE">
10 //     Copyright © 2018
11 // </copyright>
12 // <summary></summary>
13 // *****
14
15 using DataStorage;
16 using Microsoft.Win32;
17 using NUnit.Framework;
18 using OpenQA.Selenium;
19 using OpenQA.Selenium.Interactions;
20 using OpenQA.Selenium.Remote;
21 using OpenQA.Selenium.Support.Extensions;
22 using OpenQA.Selenium.Support.UI;
23 using System;
24 using System.Collections.Generic;
25 using System.Diagnostics;
26 using System.IO;
27 using System.Management;
28 using System.Text;
29 using System.Threading;
30 using System.Windows.Forms;
31 using System.Xml;
32 using System.Xml.Linq;
33 using Keys = OpenQA.Selenium.Keys;
34
35 namespace GeneralFunctionality
36 {
37     /// <summary>
38     ///     Class Functions.
39     /// </summary>
40     /// <autogeneratedoc />
41     /// TODO Edit XML Comment Template for Functions
42     public class Functions
43     {
44         /// <summary>
45         ///     The credentials
46         /// </summary>
47         /// <autogeneratedoc />
48         /// TODO Edit XML Comment Template for credentials
49         public static Credentials credentials;
50
51         /// <summary>
52         ///     The database connection
53         /// </summary>
54         /// <autogeneratedoc />
55         /// TODO Edit XML Comment Template for databaseConnection
56         public static DatabaseConnection databaseConnection;
```

```
57
58     /// <summary>
59     ///     The serial key XML
60     /// </summary>
61     /// <autogeneratedoc />
62     /// TODO Edit XML Comment Template for serialKeyXml
63     public static SerialKeyXML serialKeyXml;
64
65     /// <summary>
66     ///     The testrun
67     /// </summary>
68     /// <autogeneratedoc />
69     /// TODO Edit XML Comment Template for _testrun
70     private static string _testrun;
71
72
73     public static string _project;
74
75     /// <summary>
76     ///     The application name
77     /// </summary>
78     /// <autogeneratedoc />
79     /// TODO Edit XML Comment Template for applicationName
80     private static string applicationName;
81
82     /// <summary>
83     ///     The class name
84     /// </summary>
85     /// <autogeneratedoc />
86     /// TODO Edit XML Comment Template for className
87     private static string className;
88
89     /// <summary>
90     ///     The counter check
91     /// </summary>
92     /// <autogeneratedoc />
93     /// TODO Edit XML Comment Template for counterCheck
94     private static int counterCheck;
95
96     /// <summary>
97     ///     The testnr
98     /// </summary>
99     /// <autogeneratedoc />
100    /// TODO Edit XML Comment Template for testnr
101    private static string testnr;
102
103    public static bool uitgevoerd { get; set; }
104
105    /// <summary>
106    ///     Gets or sets the HTML page.
107    /// </summary>
108    /// <value>The HTML page.</value>
109    /// <autogeneratedoc />
110    /// TODO Edit XML Comment Template for HtmlPage
111    public static string HtmlPage { get; set; }
112
```

```

113     /// <summary>
114     ///     Gets or sets the name of the screenshot.
115     /// </summary>
116     /// <value>The name of the screenshot.</value>
117     /// <autogeneratedoc />
118     /// TODO Edit XML Comment Template for ScreenshotName
119     public static string ScreenshotName { get; set; }
120
121     /// <summary>
122     ///     Gets or sets the set testnr.
123     /// </summary>
124     /// <value>The set testnr.</value>
125     /// <autogeneratedoc />
126     /// TODO Edit XML Comment Template for SetTestnr
127     private static string SetTestnr { get; set; }
128
129     /// <summary>
130     ///     Checks the by.
131     /// </summary>
132     /// <param name="driver">The driver.</param>
133     /// <param name="idString">The identifier string.</param>
134     /// <param name="checkedBy">The checked by.</param>
135     /// <param name="testnr">The testnr.</param>
136     /// <param name="browserid">The browserid.</param>
137     /// <param name="tekst">The tekst.</param>
138     /// <param name="action">The action.</param>
139     /// <param name="url">The URL.</param>
140     /// <param name="switch1">The switch1.</param>
141     /// <param name="description">The description.</param>
142     /// <param name="inverseString">The inverse string.</param>
143     /// <param name="comment">The comment.</param>
144     /// <param name="testname">The testname.</param>
145     /// <param name="testcase">The testcase.</param>
146     /// <param name="version">The version.</param>
147     /// <param name="counter">The counter.</param>
148     /// <param name="machinenumber">The machinenumber.</param>
149     /// <param name="testelementname">The testelementname.</param>
150     /// <param name="start_url">The start URL.</param>
151     /// <param name="texttext">The texttext.</param>
152     /// <param name="testpage">The testpage.</param>
153     /// <param name="testscenario_name">Name of the testscenario.</param>
154     /// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>
155     public static bool CheckBy(
156         IWebDriver driver,
157         string idString,
158         string checkedBy,
159         string testnr = "1",
160         string browserid = "1",
161         string tekst = "",
162         string action = "",
163         string url = "",
164         string switch1 = "",
165         string description = "",
166         string inverseString = "",
167         string comment = "",
168         string testname = "",

```

```
169         string testcase = "",
170         string version = "",
171         int counter = 0,
172         string machinenumber = "",
173         string testelementname = "",
174         string start_url = "",
175         string texttext = "",
176         string testpage = "",
177         string testscenario_name = "",
178         bool opslaan = true,
179         bool strikt = true)
180     {
181         bool present;
182         var opslaan1 = opslaan;
183
184         present = true;
185
186         if (checkedBy != string.Empty || idString != string.Empty)
187         {
188             var showBorderWidth = OverallSettings.ShowBorderWidth;
189             var showShowTime = OverallSettings.ShowDuration;
190             var showBorderColor = OverallSettings.ShowBorderColor;
191             var extraTimes = OverallSettings.ShowExtraChecks;
192
193             SetTestnr = testnr;
194             ScreenshotName = string.Empty;
195             var gewoon = 3;
196             var ongewoon = 4;
197             if (inverseString == "yes")
198             {
199                 gewoon = 4;
200                 ongewoon = 3;
201             }
202
203             var result = ongewoon;
204
205             try
206             {
207                 IWebElement content = null;
208                 var actions = new Actions(driver);
209
210                 switch (checkedBy.ToUpper())
211                 {
212                     case "NAME":
213                         driver.FindElement(By.Name(idString));
214                         content = driver.FindElement(By.Name(idString));
215                         break;
216                     case "XPATH":
217                         driver.FindElement(By.XPath(idString));
218                         content = driver.FindElement(By.XPath(idString));
219                         break;
220                     case "CLASS":
221                         driver.FindElement(By.ClassName(idString));
222                         content =
223                             driver.FindElement(By.ClassName(idString));
224                         break;
```

```
225         case "ID":
226             driver.FindElement(By.Id(idString));
227             content = driver.FindElement(By.Id(idString));
228             break;
229         case "CSSSELECTOR":
230             driver.FindElement(By.CssSelector(idString));
231             content =
232                 driver.FindElement(By.CssSelector(idString));
233             break;
234     }
235
236     if (content != null && tekst != string.Empty &&
237         content.Text != string.Empty)
238     {
239         if (strikt)
240         {
241             if (tekst != string.Empty &&
242                 content.Text == tekst)
243                 result = gewoon;
244         }
245         else
246         {
247             if (tekst != string.Empty &&
248                 content.Text.IndexOf(tekst) != -1)
249                 result = gewoon;
250         }
251
252         /*if (tekst != string.Empty &&
253             content.Text != tekst)
254             result = ongewoon;*/
255     }
256
257     /*
258     if (content != null && tekst != string.Empty &&
259         content.TagName == "input" &&
260         content.GetAttribute("value") != string.Empty)
261
262         //if (content != null && tekst != string.Empty &&
263         //(content.TagName == "input" || content.TagName ==
264         "textarea") &&
265         //content.GetAttribute("value") != string.Empty)
266     {
267         if (strikt == true)
268         {
269             if (tekst != string.Empty &&
270                 content.GetAttribute("value").Trim() ==
271                 tekst)
272                 result = gewoon;
273         }
274         else
275         {
276             if (tekst != string.Empty &&
277                 content.GetAttribute("value").Trim().IndexOf
278                 (tekst) != -1)
279                 result = gewoon;
280         }
281     }
282     */
```

```
278
279             if (tekst != string.Empty &&
280                 content.GetAttribute("value").Trim() != tekst)
281                 result = ongewoon;
282         }
283     */
284
285
286     /*if (content != null && content.Text != null &&
287         content.Text != string.Empty &&
288         tekst != string.Empty)
289     {
290         // MessageBox.Show("'" + content.Text.Trim() + "'" + " +
291         tekst.Trim() + "'");
292
293         if (tekst != string.Empty &&
294             content.Text.Trim() == tekst.Trim())
295             result = gewoon;
296
297         if (tekst != string.Empty &&
298             content.Text.Trim() != tekst.Trim())
299             result = ongewoon;
300     }*/
301
302     if (tekst == string.Empty) result = gewoon;
303
304     HighlightAndScreenshot(driver, content, showShowTime,
305                             showBorderColor, showBorderWidth);
306
307     if (idString.ToUpper().Trim() != "POPUP" &&
308         idString.Trim().Length > 0)
309         if (result == ongewoon)
310             TakeScreenshot(driver, ongewoon);
311
312     present = true;
313 }
314 catch (NoSuchElementException e)
315 {
316     if (gewoon == 3)
317     {
318         present = false;
319         TakeScreenshot(driver, result);
320
321         if (counter < extraTimes)
322         {
323             counter++;
324             CheckBy(
325                 driver,
326                 idString,
327                 checkedBy,
328                 testnr,
329                 browserid,
330                 tekst,
331                 action,
332                 url,
```

```
333         switch1,
334         description,
335         version,
336         inverseString,
337         comment,
338         testname,
339         testcase,
340         counter,
341         machinenumber,
342         testelementname,
343         start_url,
344         texttext,
345         testpage,
346         testscenario_name,
347         opslaan1,
348         strikt);
349     }
350 }
351 else
352 {
353     present = true;
354 }
355 }
356 finally
357 {
358     applicationName = testname;
359     if (result == gewoon) ScreenshotName = string.Empty;
360     if (idString.ToUpper().Trim() == "POPUP") result = ↗
        gewoon;
361
362     if (opslaan)
363         TestResults.AddTestResult(
364             applicationName,
365             className,
366             _testrun,
367             Functions.testnr,
368             idString,
369             checkedBy,
370             result.ToString(),
371             ScreenshotName,
372             action,
373             tekst,
374             url,
375             switch1,
376             description,
377             browserid,
378             inverseString,
379             comment,
380             version,
381             machinenumber,
382             testelementname,
383             start_url,
384             texttext,
385             "",
386             testpage,
387             Convert.ToString(getProjectID())
```

```

388         );
389     }
390 }
391
392     return present;
393 }
394
395     /// <summary>
396     ///     Gets the HDD serial number.
397     /// </summary>
398     /// <returns>System.String.</returns>
399     /// <autogeneratedoc />
400     /// TODO Edit XML Comment Template for getHddSerialNumber
401     public static string getHddSerialNumber()
402     {
403         // --- Win32 Disk
404         var searcher = new ManagementObjectSearcher("\\root\\cimv2",
405             "select * from Win32_DiskPartition WHERE
406                 BootPartition=True");
407
408         uint diskIndex = 999;
409         foreach (ManagementObject partition in searcher.Get())
410         {
411             diskIndex =
412                 Convert.ToUInt32(partition.GetPropertyValue("Index"));
413             break; // TODO: might not be correct. Was : Exit For
414         }
415
416         // I haven't found the bootable partition. Fail.
417         if (diskIndex == 999)
418             return string.Empty;
419
420         // --- Win32 Disk Drive
421         searcher = new ManagementObjectSearcher(
422             "SELECT * FROM Win32_DiskDrive where Index = " + diskIndex);
423
424         var deviceName = "";
425         foreach (ManagementObject wmi_HD in searcher.Get())
426         {
427             deviceName = wmi_HD.GetPropertyValue("Name").ToString();
428             break; // TODO: might not be correct. Was : Exit For
429         }
430
431         // I haven't found the disk drive. Fail
432         if (string.IsNullOrEmpty(deviceName.Trim()))
433             return string.Empty;
434
435         // -- Some problems in query parsing with backslash. Using like
436         // operator
437         if (deviceName.StartsWith("\\\\.\\\\"))
438             deviceName = deviceName.Replace("\\\\.\\", "%");
439
440         // --- Physical Media
441         searcher = new ManagementObjectSearcher(
442             "SELECT * FROM Win32_PhysicalMedia WHERE Tag like '" +
443             deviceName +

```



```

442         """);
443         var serial = string.Empty;
444         foreach (ManagementObject wmi_HD in searcher.Get())
445         {
446             serial = wmi_HD.GetPropertyValue("SerialNumber").ToString();
447             break; // TODO: might not be correct. Was : Exit For
448         }
449
450         return serial;
451     }
452
453
454     public static int getMachineCode()
455     {
456         var searcher =
457             new ManagementObjectSearcher("select * from Win32_Processor");
458         var collectedInfo = "";
459         // here we will put the informa
460         foreach (ManagementObject share in searcher.Get())
461             // first of all, the processorid
462             collectedInfo += share.GetPropertyValue("ProcessorId");
463
464         searcher.Query = new ObjectQuery("select * from Win32_BIOS");
465         foreach (ManagementObject share in searcher.Get())
466             //then, the serial number of BIOS
467             collectedInfo += share.GetPropertyValue("SerialNumber");
468
469         searcher.Query = new ObjectQuery("select * from Win32_BaseBoard");
470         foreach (ManagementObject share in searcher.Get())
471             //finally, the serial number of motherboard
472             collectedInfo += share.GetPropertyValue("SerialNumber");
473
474         // patch luca bernardini
475         if (string.IsNullOrEmpty(collectedInfo) | (collectedInfo == "00") |
476             (collectedInfo.Length <= 3))
477             collectedInfo += getHddSerialNumber();
478
479         return getEightByteHash(collectedInfo, 100000);
480     }
481
482     /// <summary>
483     ///     Gets the eight byte hash.
484     /// </summary>
485     /// <param name="s">The s.</param>
486     /// <param name="MUST_BE_LESS_THAN">The must be less than.</param>
487     /// <returns>System.Int32.</returns>
488     /// <autogeneratedoc />
489     /// TODO Edit XML Comment Template for getEightByteHash
490     public static int getEightByteHash(string s,
491         int MUST_BE_LESS_THAN = 1000000000)
492     {
493         //This function generates a eight byte hash
494

```

```
495         //The length of the result might be changed to any length
496         //just set the amount of zeroes in MUST_BE_LESS_THAN
497         //to any length you want
498         uint hash = 0;
499
500         foreach (var b in Encoding.Unicode.GetBytes(s))
501         {
502             hash += b;
503             hash += hash << 10;
504             hash ^= hash >> 6;
505         }
506
507         hash += hash << 3;
508         hash ^= hash >> 11;
509         hash += hash << 15;
510
511         var result = (int)(hash % MUST_BE_LESS_THAN);
512         var check = MUST_BE_LESS_THAN / result;
513
514         if (check > 1) result *= check;
515
516         return result;
517     }
518
519
520     /// <summary>
521     ///     Checks the screenshot dir.
522     /// </summary>
523     /// <param name="dirName">Name of the dir.</param>
524     /// <autogeneratedoc />
525     /// TODO Edit XML Comment Template for CheckScreenshotDir
526     public static void CheckScreenshotDir(string dirName = "screenshots")
527     {
528         var screenshotsPath = GetCurrentDir(1);
529         var pathG = screenshotsPath.Split('\\');
530         var g = string.Empty;
531
532         for (var i = 0; i < pathG.Length - 2; i++) g += pathG[i] + "\\";
533
534         screenshotsPath = g;
535         var di = new DirectoryInfo(screenshotsPath);
536         var dis = new DirectoryInfo(screenshotsPath + "\\" + dirName);
537         var diserror = new DirectoryInfo(screenshotsPath + "\\" + dirName);
538         if (!dis.Exists)
539         {
540             var driverPathSub = di.CreateSubdirectory(dirName);
541             var applicationPathSub =
542                 di.CreateSubdirectory(dirName + "\\" +
543                                     applicationName.Trim());
544             var applicationPathSubErrors =
545                 di.CreateSubdirectory(dirName + "\\" +
546                                     applicationName.Trim());
547         }
548         else
549         {
```

```
550         var applicationPath =
551             new DirectoryInfo(screenshotsPath + "\\\" + dirName + "\\\" ↗
552                 +
553                 applicationName.Trim());
554         if (applicationPath.Exists)
555         {
556         }
557         else
558         {
559             var applicationPathSub =
560                 di.CreateSubdirectory(dirName + "\\\" +
561                 applicationName.Trim());
562         }
563         applicationPath = new DirectoryInfo(screenshotsPath + "\\\" +
564             dirName + "\\\" +
565             applicationName.Trim());
566
567         if (!applicationPath.Exists)
568         {
569             var applicationPathSubErrors =
570                 di.CreateSubdirectory(dirName + "\\\" +
571                 applicationName.Trim());
572         }
573     }
574 }
575
576 public static void CheckDir(string project)
577 {
578     var projectPath = GetCurrentDir(1) + project + @"\CSV";
579
580     var pathG = projectPath.Split('\\');
581     var g = string.Empty;
582
583     for (var i = 0; i < pathG.Length - 1; i++) g += pathG[i] + "\\\";
584
585     projectPath = g;
586     var di = new DirectoryInfo(projectPath);
587     var dis = new DirectoryInfo(projectPath + @"\CSV");
588     var diserror = new DirectoryInfo(projectPath + @"\CSV");
589     if (!dis.Exists)
590     {
591         var driverPathSub = di.CreateSubdirectory("CSV");
592         var applicationPathSub = di.CreateSubdirectory("CSV\\");
593         var applicationPathSubErrors = di.CreateSubdirectory("CSV\ ↗
594             \");
595     }
596     else
597     {
598         var applicationPath =
599             new DirectoryInfo(projectPath + @"\CSV");
600         if (applicationPath.Exists)
601         {
602         }
603         else
604         {
605         }
```

```

604         var applicationPathSub = di.CreateSubdirectory("\\CSV");
605     }
606
607     applicationPath = new DirectoryInfo(projectPath + "\\CSV");
608
609     if (!applicationPath.Exists)
610     {
611         var applicationPathSubErrors =
612             di.CreateSubdirectory("CSV\\");
613     }
614 }
615
616
617
618 /// <summary>
619 ///     Creates the HTML.
620 /// </summary>
621 /// <param name="testrun">The testrun.</param>
622 /// <autogeneratedoc />
623 /// TODO Edit XML Comment Template for CreateHTML
624 public static void CreateHTML(string testrun)
625 {
626     try
627     {
628         var totalTestResultsHtmls =
629             new List<TestResults.TotalTestResultsHTML>();
630         totalTestResultsHtmls =
631             TestResults.GetTotalTestResultsHTML(testrun);
632
633         var testResults = new List<TestResults.TestResultsHTML>();
634         testResults = TestResults.GetTestResultsHTML(testrun);
635         setApplicationName(testResults[0].application);
636
637         var timestring = "run_" +
638             string.Format("{0,8:D8}",
639                 Convert.ToInt32(testrun));
640         var message = "<!DOCTYPE html>\r\n<html>\r\n<head>";
641         message += "\t<meta charset=\"UTF-8\">\r\n";
642         message +=
643             "\t<meta name=\"viewport\" content=\"width=device-
644             width,initial-scale=1\">\r\n";
645         message += "\t<title>Testrun: " + testResults[0].application
646             +
647             " _" + timestring + "</title>\r\n";
648         message += "\t<link rel=\"stylesheet\" href=\"\" +
649             GetCurrentDir(0) + "\\style.css\">\r\n";
650         message += "</head>";
651         message += "\t<body class=\"dark-theme\"><script id=\"__bs_script__
652             \>/*! [CDATA[document.write(\"<script async src='/'
653             browser-sync/browser-sync-client.js?v=2.23.6'><\\script>
654             \".replace(\"HOST\", location.hostname));//]]></script>\r
655             \n";
656         message += "<div id=\"wrapper\">\r\n";
657         message += "\t<h1>Test: " + testResults[0].application.Trim()
658             +

```

```

653         "</h1>\r\n";
654         message += "\t<h2>Testrun: " +
655             testResults[0].testrun.ToString().Trim() +
656             "</h2>\r\n";
657         message += "\t<h2>Browser: " +
658             totalTestResultsHtmls[0].testrun_browser +
659             "</h2>\r\n";
660         message += "\t<div id=\"dashboard\">\r\n";
661
662         // TestResults total
663         var passedInt =
664             Convert.ToDouble(totalTestResultsHtmls
665                 [0].testrun_passed);
666         var failedInt =
667             Convert.ToDouble(totalTestResultsHtmls
668                 [0].testrun_failed);
669         var passedAfgerond =
670             Convert.ToInt32((passedInt / (passedInt + failedInt) * 100 -
671                 0.50));
672         var passedAfgerond1 = string.Format("{0:F1}",
673             passedInt / (passedInt + failedInt) * 100);
674         message += "\t\t<div class=\"row\">\r\n";
675
676         // Grafic
677         message += "\t\t\t<div class=\"col-4\">\r\n";
678         message += "\t\t\t\t<div class=\"card\">\r\n";
679         message +=
680             "\t\t\t\t\t<div class=\"card__content row donut\">\r\n";
681         message +=
682             "\t\t\t\t\t\t<div id=\"chart\" class=\"col-5\" data-
683                 percent=\"\" +
684                 passedAfgerond1
685                 + "\">\r\n";
686         message +=
687             "\t\t\t\t\t\t\t<svg width=\"100%\" height=\"100%\"
688                 viewBox=\"0 0 40 25\" class=\"donut\">\r\n";
689         message +=
690             "\t\t\t\t\t\t\t\t<circle class=\"donut-ring\" cx=\"20\"
691                 cy=\"13\" r=\"16\" fill=\"transparent\"></circle>\r\n";
692         message +=
693             "\t\t\t\t\t\t\t\t\t<circle class=\"donut-segment p-\"
694                 +
695                 passedAfgerond
696                 + "\" cx=\"20\" cy=\"13\" r=\"16\" fill=
697                 \"transparent\" stroke-dashoffset=\"25\"></circle>\r\n";
698         message += "\t\t\t\t\t\t</svg>\r\n";
699         message += "\t\t\t\t</div>\r\n";
700         message += "\t\t\t<div class=\"col-7\">\r\n";
701         message += "\t\t\t\t<ul id=\"runtime\">\r\n";
702         message +=
703             "\t\t\t\t\t<li><strong>Start time: </strong></br>\"
704                 +
705                 string.Format(
706                     \"{0:dd/MM/yy HH:mm:ss}\",
707                     totalTestResultsHtmls[0].testrun_begintime) +
708                 "</li>\r\n";

```

```

700         message +=
701             "\t\t\t\t\t\t\t\t<li><strong>End Time: </strong></br>" +
702             string.Format(
703                 "{0:dd/MM/yy HH:mm:ss}",
704                 totalTestResultsHtmls[0].testrun_endtime) + "</li>\r\n";
705
706         message +=
707             "\t\t\t\t\t\t\t\t<li><strong>Execution Time: </strong></br>"
708             + string.Format("{0:HH:mm:ss}",
709                 totalTestResultsHtmls[0].testrun_time) + "</li>\r\n";
710         message += "\t\t\t\t\t\t\t\t</ul>\r\n";
711         message += "\t\t\t\t\t\t\t\t</div>\r\n";
712         message += "\t\t\t\t\t\t\t\t</div>\r\n";
713         message += "\t\t\t\t\t\t\t\t</div>\r\n";
714
715         // Passed
716         message += "\t\t\t\t<div class=\"col-4\">\r\n";
717         message += "\t\t\t\t<div class=\"card success\">\r\n";
718         message += "\t\t\t\t\t\t<div class=\"card__content\">\r\n";
719         message += "\t\t\t\t\t\t\t<h3>Passed</h3>\r\n";
720         message += "\t\t\t\t\t\t\t<div class=\"counter\">\r\n";
721         message += "\t\t\t\t\t\t\t\t<div>" + passedInt + "</div>\r\n";
722         message += "\t\t\t\t\t\t\t\t<svg class=\"check\">\r\n";
723         message +=
724             "\t\t\t\t\t\t\t\t\t\t<line x1=\"5\" y1=\"15\" x2=\"15\" y2=
725             \"25\" />\r\n";
726         message +=
727             "\t\t\t\t\t\t\t\t\t\t<line x1=\"13\" y1=\"25\" x2=\"30\" y2=
728             \"5\" />\r\n";
729         message += "\t\t\t\t\t\t\t\t</div>\r\n";
730         message += "\t\t\t\t\t\t\t\t</div>\r\n";
731         message += "\t\t\t\t\t\t\t\t</div>\r\n";
732
733         // Failed
734         message += "\t\t\t\t<div class=\"col-4\">\r\n";
735         message += "\t\t\t\t<div class=\"card error\">\r\n";
736         message += "\t\t\t\t\t\t<div class=\"card__content\">\r\n";
737         message += "\t\t\t\t\t\t\t<h3>Failed</h3>\r\n";
738         message += "\t\t\t\t\t\t\t<div class=\"counter\">\r\n";
739         message += "\t\t\t\t\t\t\t\t<div>" + failedInt + "</div>\r\n";
740         message += "\t\t\t\t\t\t\t\t<svg class=\"cross\">\r\n";
741         message +=
742             "\t\t\t\t\t\t\t\t\t\t<line x1=\"1\" y1=\"25\" x2=\"25\" y2=
743             \"1\" />\r\n";
744         message +=
745             "\t\t\t\t\t\t\t\t\t\t<line x1=\"1\" y1=\"1\" x2=\"25\" y2=
746             \"25\" />\r\n";
747         message += "\t\t\t\t\t\t\t\t</div>\r\n";
748         message += "\t\t\t\t\t\t\t\t</div>\r\n";
749         message += "\t\t\t\t\t\t\t\t</div>\r\n";

```

```

750         message += "\t\t</div>\r\n";
751
752         // Header details
753         message += "\t\t<div class=\"list\">\r\n";
754         message += "\t\t\t<div class=\"list__content\">\r\n";
755         message += "\t\t\t\t<div class=\"list__content__row\">\r\n";
756         message += "\t\t\t\t\t<div class=\"row details\">\r\n";
757         message += "\t\t\t\t\t\t<div class=\"col-1\">\r\n";
758         message +=
759             "\t\t\t\t\t\t\t<span><strong>Nr.</strong></span>\r\n";
760         message += "\t\t\t\t\t\t\t</div>\r\n";
761         message += "\t\t\t\t\t\t\t<div class=\"col-3 action\">\r\n";
762         message +=
763             "\t\t\t\t\t\t\t\t<span><strong>attribute</strong></span>\r\n";
764         message += "\t\t\t\t\t\t\t\t</div>\r\n";
765         message += "\t\t\t\t\t\t\t</div>\r\n";
766         message +=
767             "\t\t\t\t\t\t\t<div class=\"element\"><strong>element name</strong>\r\n";
768         message += "\t\t\t\t\t\t\t</div>\r\n";
769         message += "\t\t\t\t\t\t\t</div>\r\n";
770
771         var showErrorDisplay =
772             OverallSettings.ShowErrorDisplay.ToString();
773
774         // Details
775         foreach (var testResult in testResults)
776         {
777             var testOK = "list__content__row success";
778             if (testResult.result == "4")
779                 testOK = "list__content__row error";
780
781             if (showErrorDisplay == "0" && testResult.result == "4")
782             {
783                 message += "\t\t\t\t\t<div class=\"" + testOK + "\">\r\n";
784                 message += "\t\t\t\t\t\t<div class=\"row details\">\r\n";
785                 message += "\t\t\t\t\t\t\t<div class=\"col-1\">\r\n";
786                 message += "\t\t\t\t\t\t\t\t<span>" + testResult.testnr
787                     +
788                     "</span>\r\n";
789                 message += "\t\t\t\t\t\t\t\t</div>\r\n";
790                 message +=
791                     "\t\t\t\t\t\t\t\t\t<div class=\"col-3 action\">\r\n";
792                 message += "\t\t\t\t\t\t\t\t\t\t<span>" +
793                     testResult.attribute + "</span>\r\n";
794                 message += "\t\t\t\t\t\t\t\t\t\t</div>\r\n";
795                 message += "\t\t\t\t\t\t\t\t\t\t</div>\r\n";
796                 message += "\t\t\t\t\t\t\t\t\t\t<div class=\"element\">" +
797                     testResult.elementname + "(" +
798                     testResult.element + "></div>\r\n";
799                 message += "\t\t\t\t\t\t\t\t\t\t</div>\r\n";
800             }

```

```

801
802         if (showErrorDisplay == "1")
803         {
804             message += "\t\t\t\t<div class=\"" + testOK + "\">\r\n";
805             message += "\t\t\t\t\t\t<div class=\"row details\">\r\n";
806             message += "\t\t\t\t\t\t\t\t<div class=\"col-1\">\r\n";
807             message += "\t\t\t\t\t\t\t\t\t\t<span>" + testResult.testnr
808             +
809             "\t\t\t\t\t\t\t\t\t\t\t\t</span>\r\n";
810             message += "\t\t\t\t\t\t\t\t\t\t\t\t</div>\r\n";
811             message += "\t\t\t\t\t\t\t\t\t\t\t\t\t\t<div class=\"col-3 action\">\r\n";
812             message += "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t<span>" +
813             testResult.attribute + "</span>\r\n";
814             message += "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t</div>\r\n";
815             message += "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t</div>\r\n";
816             message +=
817             "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t<div class=\"element\">" +
818             testResult.elementname + "(" +
819             testResult.element + ">\r\n";
820             message += "\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t</div>\r\n";
821         }
822     }
823
824     message += "\t\t\t\t</div>\r\n";
825     message += "\t\t</div>\r\n";
826     message += "\t</div>\r\n";
827     message += "</body>\r\n";
828     message += "</html>";
829     TestReportToFile(message, timestring);
830 }
831 catch (Exception e)
832 {
833     Console.WriteLine(e);
834     MessageBox.Show(e.Message);
835 }
836 }
837
838 /// <summary>
839 ///     Deteles the images.
840 /// </summary>
841 /// <autogeneratedoc />
842 /// TODO Edit XML Comment Template for DeteleImages
843 public static void DeteleImages()
844 {
845     var screenshotsPath = GetCurrentDir(1);
846     var di = new DirectoryInfo(screenshotsPath + "screenshots\\" +
847         applicationName.Trim());
848     foreach (var file in di.GetFiles()) file.Delete();
849 }
850
851 /// <summary>
852 ///     Gets the counter check.
853 /// </summary>

```



```
854     /// <returns>System.Int32.</returns>
855     /// <autogeneratedoc />
856     /// TODO Edit XML Comment Template for GetCounterCheck
857     public static int GetCounterCheck()
858     {
859         return counterCheck;
860     }
861
862     /// <summary>
863     ///     Gets the credentials.
864     /// </summary>
865     /// <param name="bestandsnaam">The bestandsnaam.</param>
866     /// <returns>Credentials.</returns>
867     /// <autogeneratedoc />
868     /// TODO Edit XML Comment Template for GetCredentials
869     public static Credentials GetCredentials(string bestandsnaam)
870     {
871         var credentials = new Credentials(bestandsnaam);
872         return credentials;
873     }
874
875     /// <summary>
876     ///     Gets the current dir.
877     /// </summary>
878     /// <param name="niveau">The niveau.</param>
879     /// <returns>System.String.</returns>
880     /// <autogeneratedoc />
881     /// TODO Edit XML Comment Template for GetCurrentDir
882     public static string GetCurrentDir(int niveau)
883     {
884         var ValueList = new List<string>();
885         if (File.Exists("settings.xml"))
886         {
887             var xDoc = new XmlDocument();
888             xDoc.Load("settings.xml");
889             try
890             {
891                 var xelement = XElement.Load("settings.xml");
892                 var elementVars = xelement.Elements();
893                 foreach (var elementVar in elementVars)
894                     ValueList.Add(elementVar.Value);
895             }
896             catch (Exception ex)
897             {
898                 General.LogMessage(ex.Message + ex.StackTrace, 4);
899                 MessageBox.Show(
900                     "The 'settings.xml' file is not found!!!\r\nPlease
901                     fill in the correct drive and directory.\r\n"
902                     + Environment.CurrentDirectory,
903                     "Settings Error",
904                     MessageBoxButtons.OK,
905                     MessageBoxIcon.Error);
906                 ValueList[0] = Environment.CurrentDirectory;
907             }
908         }
909         else
```

```

909         {
910             MessageBox.Show(
911                 "The 'settings.xml' file is not found!!!\r\nPlease fill
912                 in the correct drive and directory.\r\n"
913                 + Environment.CurrentDirectory,
914                 "Settings Error",
915                 MessageBoxButtons.OK,
916                 MessageBoxIcon.Error);
917         }
918         return ValueList[niveau];
919     }
920
921     /// <summary>
922     ///     Gets the directory.
923     /// </summary>
924     /// <returns>System.String.</returns>
925     /// <autogeneratedoc />
926     /// TODO Edit XML Comment Template for GetDirectory
927     public static string GetDirectory()
928     {
929         var driverPath = Environment.CurrentDirectory;
930
931         var directory = driverPath.Split('\\');
932         var intX = 0;
933         foreach (var dir in directory)
934             if (dir == "SeleniumCSharpTests")
935                 break;
936             else
937                 intX++;
938
939         var di = string.Empty;
940         for (var x = 0; x < directory.Length - intX - 1; x++)
941             di += directory[x] + "\\";
942
943         return di;
944     }
945
946     /// <summary>
947     ///     Gets the settings XML strings.
948     /// </summary>
949     /// <param name="xmlFile">The XML file.</param>
950     /// <param name="dirnr">The dirnr.</param>
951     /// <returns>List<System.String>.</returns>
952     public static List<string> GetSettingsXmlStrings(string xmlFile,
953         int dirnr = 0)
954     {
955         var ValueList = new List<string>();
956
957         var di = GetCurrentDir(dirnr);
958         if (File.Exists(di + xmlFile + ".xml"))
959         {
960             var xDoc = new XmlDocument();
961             xDoc.Load(di + xmlFile + ".xml");
962             try
963             {

```

```

964         var xelement = XElement.Load(di + xmlFile + ".xml");
965         var elementVars = xelement.Elements().Elements();
966         foreach (var elementVar in elementVars)
967             ValueList.Add(elementVar.Value);
968     }
969     catch (Exception ex)
970     {
971         General.LogMessage(ex.Message + ex.StackTrace, 4);
972         MessageBox.Show(
973             "The '" + di + xmlFile + ".xml"
974             + ".xml' file is not found!!!\r\nPlease fill in the  ➤
correct drive and directory.",
975             "Settings Error",
976             MessageBoxButtons.OK,
977             MessageBoxIcon.Error);
978     }
979 }
980 else
981 {
982     if (xmlFile != "serialXML")
983         MessageBox.Show(
984             "The '" + di + xmlFile
985             + ".xml' file is not found!!!\r\nPlease fill in the  ➤
correct drive and directory.",
986             "Settings Error",
987             MessageBoxButtons.OK,
988             MessageBoxIcon.Error);
989 }
990
991 return ValueList;
992 }
993
994 /// <summary>
995 ///     Haals the elementop.
996 /// </summary>
997 /// <param name="driver">The driver.</param>
998 /// <param name="testElement">The test element.</param>
999 /// <param name="testAttribute">The test attribute.</param>
1000 /// <returns>IWebElement.</returns>
1001 /// <autogeneratedoc />
1002 /// TODO Edit XML Comment Template for HaalElementop
1003 public static IWebElement HaalElementop(IWebDriver driver,
1004     string testElement, string testAttribute)
1005 {
1006     IWebElement webElement = null;
1007
1008     switch (testAttribute)
1009     {
1010         case "ID":
1011             // select the drop down list
1012             webElement = driver.FindElement(By.Id(testElement));
1013             break;
1014         case "NAME":
1015             webElement = driver.FindElement(By.Name(testElement));
1016             break;
1017         case "CLASS":

```

```
1018         webElement = driver.FindElement(By.ClassName
1019         (testElement));
1020         break;
1021     case "XPATH":
1022         webElement = driver.FindElement(By.XPath(testElement));
1023         break;
1024     case "CSSSELECTOR":
1025         webElement =
1026             driver.FindElement(By.CssSelector(testElement));
1027         break;
1028     }
1029     return webElement;
1030 }
1031
1032 public static bool IsElementVisible(IWebElement element)
1033 {
1034     return element.Displayed && element.Enabled;
1035 }
1036
1037 /// <summary>
1038 ///     Highlights the and screenshot.
1039 /// </summary>
1040 /// <param name="driver1">The driver1.</param>
1041 /// <param name="context">The context.</param>
1042 /// <param name="sleepTime">The sleep time.</param>
1043 /// <param name="borderColor">Color of the border.</param>
1044 /// <param name="borderWidth">Width of the border.</param>
1045 /// <autogeneratedoc />
1046 /// TODO Edit XML Comment Template for HighlightAndScreenshot
1047 public static void HighlightAndScreenshot(
1048     IWebDriver driver1,
1049     IWebElement context,
1050     int sleepTime,
1051     string borderColor,
1052     int borderWidth)
1053 {
1054     var oldScript = context.GetAttribute("style");
1055     var rc = (RemoteWebElement)context;
1056     var actions = new Actions(driver1);
1057     if (rc.Displayed)
1058     {
1059         var visible = IsElementVisible(context);
1060         if (visible)
1061         {
1062             actions = new Actions(driver1);
1063             actions.MoveToElement(context);
1064             actions.Perform();
1065         }
1066
1067         var driver = (IJavaScriptExecutor)rc.WrappedDriver;
1068
1069         var script = "arguments[0].style.cssText = \"border-width: "
1070             +
1071             borderWidth
1072             + "px; border-style: solid; border-color: "
```

```
1072         + borderColor.ToLower() + "\"";
1073         driver.ExecuteScript(script, rc);
1074
1075         Thread.Sleep(sleepTime);
1076
1077         var clear = string.Empty;
1078         if (oldScript.Length > 0)
1079             clear = @"arguments[0].style.cssText = 'oldScript'";
1080         else
1081             clear = @"arguments[0].style.cssText = \"\"\"; ";
1082
1083         driver.ExecuteScript(clear, rc);
1084     }
1085 }
1086
1087 /// <summary>
1088 ///     Initializes the serial key.
1089 /// </summary>
1090 /// <returns>System.Int32.</returns>
1091 public static int InitializeSerialKey()
1092 {
1093     var value = 1;
1094     var xmlSerial = GetSettingsXmlStrings("serialXML");
1095     if (xmlSerial.Count > 0)
1096     {
1097         serialKeyXml = new SerialKeyXML(
1098             xmlSerial[0],
1099             xmlSerial[1],
1100             xmlSerial[2],
1101             xmlSerial[3],
1102             xmlSerial[4],
1103             xmlSerial[5]);
1104         General.SetSerialKeyXML(serialKeyXml);
1105         value = 2;
1106     }
1107
1108     return value;
1109 }
1110
1111 /// <summary>
1112 ///     Initializes the database connection.
1113 /// </summary>
1114 /// <param name="loginU">if set to <c>true</c> [login u].</param>
1115 public static void InitializeDatabaseConnection(bool loginU)
1116 {
1117     /*
1118     * Gebruiker dient in te loggen op een andere database als waar
1119     * de data staan
1120     */
1121     //var host = "SCHMEITSNAS.SYNOLOGY.ME";
1122     //var database = "autotest";
1123     //var port = "3306";
1124     //var user = "autotest";
1125     //var password = "NiFSi1fZjbJ4zXE8";
1126     //
```

```

1127     var host = "";
1128     var database = "";
1129     var port = "";
1130     var user = "";
1131     var password = "";
1132     General.LogMessage("InitializeDatabaseConnection(" + loginU +
1133         ");",
1134         1);
1135
1136     if (loginU == false)
1137     {
1138         var xelement = XElement.Load(GetCurrentDir(0) +
1139             "connect.xml");
1140         General.LogMessage("xelement: " + xelement, 1);
1141         var testsElements = xelement.Elements();
1142         foreach (var testElement in testsElements)
1143         {
1144             host = testElement.Element("host").Value;
1145             database = testElement.Element("database").Value;
1146             port = testElement.Element("port").Value;
1147             user = testElement.Element("username").Value;
1148             password = testElement.Element("password").Value;
1149         }
1150     }
1151
1152     // anders hard gecodeerde gegevens gebruiker voor de database
1153     // connectie.
1154     databaseConnection = new DatabaseConnection(
1155         host,
1156         database,
1157         port,
1158         user,
1159         password);
1160
1161     General.LogMessage(databaseConnection.ToString(), 1);
1162     General.LogMessage("Host: " + databaseConnection.Host, 1);
1163     General.LogMessage("Database: " + databaseConnection.Database,
1164         1);
1165     General.LogMessage("Port: " + databaseConnection.Port, 1);
1166     General.LogMessage("User: " + databaseConnection.User, 1);
1167     General.LogMessage("Password: " + databaseConnection.Password,
1168         1);
1169     General.SetDatabaseConnection(databaseConnection);
1170 }
1171
1172 /// <summary>
1173 ///     Determines whether [is element present] [the specified
1174 ///     driver].
1175 /// </summary>
1176 /// <param name="driver">The driver.</param>
1177 /// <param name="idString">The identifier string.</param>
1178 /// <returns><c>true</c> if [is element present] [the specified
1179 ///     driver]; otherwise, <c>false</c>.</returns>
1180
1181 /// <autogeneratedoc />
1182 /// TODO Edit XML Comment Template for IsElementPresent
1183 public static bool IsElementPresent(IWebDriver driver, string

```

```
        idString)
    {
1176         try
1177         {
1178             driver.FindElement(By.Id(idString));
1179             return true;
1180         }
1181         catch (NoSuchElementException)
1182         {
1183             return false;
1184         }
1185     }
1186 }
1187
1188 /// <summary>
1189 ///     Determines whether this instance is windows10.
1190 /// </summary>
1191 /// <returns><c>true</c> if this instance is windows10; otherwise,  ➤
1192 ///     <c>false</c>.</returns>
1193 /// <autogeneratedoc />
1194 /// TODO Edit XML Comment Template for IsWindows10
1195 public static bool IsWindows10()
1196 {
1197     var reg =
1198         Registry.LocalMachine.OpenSubKey(
1199             @"SOFTWARE\Microsoft\Windows NT\CurrentVersion");
1200
1201     var productName = (string)reg.GetValue("ProductName");
1202
1203     return productName.StartsWith("Windows 10");
1204 }
1205
1206 public static void TestUitgevoerd(bool uit)
1207 {
1208     uitgevoerd = uit;
1209 }
1210
1211 /// <summary>
1212 ///     Sets the name of the application.
1213 /// </summary>
1214 /// <param name="name">The name.</param>
1215 /// <autogeneratedoc />
1216 /// TODO Edit XML Comment Template for setApplicationName
1217 public static void setApplicationName(string name)
1218 {
1219     applicationName = name;
1220 }
1221
1222 /// <summary>
1223 ///     Sets the name of the class.
1224 /// </summary>
1225 /// <param name="name">The name.</param>
1226 /// <autogeneratedoc />
1227 /// TODO Edit XML Comment Template for setClassName
1228 public static void setClassName(string name)
1229 {
1230     className = name;
1231 }
```

```

1230     }
1231
1232     /// <summary>
1233     ///     Sets the counter check.
1234     /// </summary>
1235     /// <param name="value">The value.</param>
1236     /// <autogeneratedoc />
1237     /// TODO Edit XML Comment Template for SetCounterCheck
1238     public static void SetCounterCheck(int value)
1239     {
1240         counterCheck = value;
1241     }
1242
1243     /// <summary>
1244     ///     Sets the HTML code.
1245     /// </summary>
1246     /// <param name="timestring">The timestring.</param>
1247     /// <returns>System.String.</returns>
1248     /// <autogeneratedoc />
1249     /// TODO Edit XML Comment Template for SetHTMLCode
1250     public static string SetHTMLCode(string timestring)
1251     {
1252         var testResults = new List<TestResults.TestResultsHTML>();
1253         testResults = TestResults.GetTestResultsHTML(timestring);
1254         setApplicationName(testResults[0].application);
1255
1256         var screenshotsPath = GetCurrentDir(1);
1257         var pathG = screenshotsPath.Split('\\');
1258         var g = string.Empty;
1259
1260         for (var i = 0; i < pathG.Length - 2; i++) g += pathG[i] + "\\";
1261
1262         screenshotsPath = g;
1263         var name1 = applicationName.Trim();
1264
1265         CheckScreenshotDir("TestReports\\" + name1);
1266
1267         var relativePath = screenshotsPath + "TestReports\\" + name1 + "\\ 
1268         \";
1269         HtmlPage = relativePath + "TestReport_" + timestring + ".html";
1270         return HtmlPage;
1271     }
1272
1273     /// <summary>
1274     ///     Sets the testnr.
1275     /// </summary>
1276     /// <param name="testnr">The testnr.</param>
1277     /// <autogeneratedoc />
1278     /// TODO Edit XML Comment Template for setTestnr
1279     public static void setTestnr(string testnr)
1280     {
1281         Functions.testnr = testnr;
1282     }
1283
1284     /// <summary>
1285     ///     Sets the testrun identifier.

```



```
1285     /// </summary>
1286     /// <param name="testrun">The testrun.</param>
1287     /// <autogeneratedoc />
1288     /// TODO Edit XML Comment Template for setTestrunID
1289     public static void setTestrunID(string testrun)
1290     {
1291         _testrun = testrun;
1292     }
1293
1294     public static void setProjectName(string project)
1295     {
1296         _project = project;
1297     }
1298
1299     public static int getProjectID()
1300     {
1301         var projectID = 0;
1302         if (_project != null)
1303         {
1304             var query =
1305                 "SELECT id FROM projects WHERE UPPER(project_name) = '" +
1306                 _project.ToUpper() + "'";
1307             var ds = GenericDataRead.GetData(query);
1308             if (ds.Rows.Count > 0)
1309                 projectID = Convert.ToInt32(ds.Rows[0][0].ToString());
1310         }
1311
1312         return projectID;
1313     }
1314
1315     /// <summary>
1316     ///     Shows the data.
1317     /// </summary>
1318     /// <param name="driver">The driver.</param>
1319     /// <autogeneratedoc />
1320     /// TODO Edit XML Comment Template for ShowData
1321     public static void ShowData(IWebDriver driver)
1322     {
1323         var PageURL = driver.Url;
1324         var PageSource = driver.PageSource;
1325
1326         var stringL = PageSource;
1327         General.LogMessage(stringL, 5);
1328     }
1329
1330     /// <summary>
1331     ///     Splits the bestand.
1332     /// </summary>
1333     /// <param name="bestand">The bestand.</param>
1334     /// <returns>System.String.</returns>
1335     /// <autogeneratedoc />
1336     /// TODO Edit XML Comment Template for SplitBestand
1337     public static string SplitBestand(string bestand)
1338     {
1339         var ele = bestand.Split('\\');
1340         var laatste = ele[ele.Length - 1];
```

```
1341         var naam = laatste.Split('.');
1342         return naam[0];
1343     }
1344
1345     /// <summary>
1346     ///     Switches to next tab.
1347     /// </summary>
1348     /// <param name="driver">The driver.</param>
1349     /// <param name="handlesNr">The handles nr.</param>
1350     /// <autogeneratedoc />
1351     /// TODO Edit XML Comment Template for SwitchToNextTab
1352     public static void SwitchToNextTab(IWebDriver driver, int handlesNr)
1353     {
1354         if (driver.WindowHandles.Count > handlesNr)
1355         {
1356             driver.SwitchTo().Window(driver.WindowHandles[handlesNr]);
1357         }
1358         else
1359         {
1360             Thread.Sleep(2000);
1361             SwitchToNextTab(driver, handlesNr);
1362         }
1363     }
1364
1365     /// <summary>
1366     ///     Takes the screenshot.
1367     /// </summary>
1368     /// <param name="driver">The driver.</param>
1369     /// <param name="result">The result.</param>
1370     /// <param name="name">The name.</param>
1371     /// <autogeneratedoc />
1372     /// TODO Edit XML Comment Template for TakeScreenshot
1373     public static void TakeScreenshot(IWebDriver driver, int result,
1374         string name = "screenshot")
1375     {
1376         var screenshotsPath = GetCurrentDir(1);
1377         var pathG = screenshotsPath.Split('\\');
1378         var g = string.Empty;
1379
1380         for (var i = 0; i < pathG.Length - 2; i++) g += pathG[i] + "\\";
1381
1382         screenshotsPath = g;
1383         var name1 = applicationName.Trim();
1384
1385         if (result == 4)
1386         {
1387             var relativePath =
1388                 screenshotsPath + "screenshots\\" + name1 + "\\";
1389             ScreenshotName =
1390                 relativePath + name1 + "-" +
1391                 string.Format("{0:yyyyMMddHHmmssFFF}", DateTime.UtcNow)
1392                 + "testnr_" + SetTestnr + ".png";
1393             General.LogMessage("Screenshot name: " + ScreenshotName, 1);
1394             driver.TakeScreenshot().SaveAsFile(ScreenshotName,
1395                 ScreenshotImageFormat.Png);
1396         }
1397     }
```

```
1397         General.LogMessage("Error", result, name1,
1398             Convert.ToInt32(SetTestnr));
1399     }
1400
1401     ShowData(driver);
1402 }
1403
1404 /// <summary>
1405 ///     Teksts the vervanging.
1406 /// </summary>
1407 /// <param name="tekst">The tekst.</param>
1408 /// <returns>System.String.</returns>
1409 /// <autogeneratedoc />
1410 /// TODO Edit XML Comment Template for TekstVervanging
1411 public static string TekstVervanging(string tekst)
1412 {
1413     var tekstGewijzigd = tekst;
1414     switch (tekst)
1415     {
1416         case "=CurrentDate()":
1417             tekstGewijzigd = DateTime.Now.Day + "-" +
1418                 DateTime.Now.Month + "-" +
1419                 DateTime.Now.Year;
1420             break;
1421     }
1422
1423     return tekstGewijzigd;
1424 }
1425
1426 /// <summary>
1427 ///     Tests the report to file.
1428 /// </summary>
1429 /// <param name="message">The message.</param>
1430 /// <param name="timestring">The timestring.</param>
1431 /// <autogeneratedoc />
1432 /// TODO Edit XML Comment Template for TestReportToFile
1433 public static void TestReportToFile(string message, string timestring)
1434 {
1435     SetHTMLCode(Convert.ToInt32(timestring.Substring(4, 8)).ToString
1436         ());
1437     var sw = File.AppendText(HtmlPage); // Change filename
1438     try
1439     {
1440         var logLine = $"{message}";
1441         sw.WriteLine(logLine);
1442     }
1443     finally
1444     {
1445         sw.Close();
1446     }
1447 }
1448
1449 public static void CodeToFile(string message, string timestring)
1450 {
1451     var sw = File.AppendText(HtmlPage); // Change filename
```

```

1451         try
1452         {
1453             var logLine = $"{message}";
1454             sw.WriteLine(logLine);
1455         }
1456         finally
1457         {
1458             sw.Close();
1459         }
1460     }
1461
1462     /// <summary>
1463     ///     Teststaps the specified driver.
1464     /// </summary>
1465     /// <param name="driver">The driver.</param>
1466     /// <param name="machinestatic">The machinestatic.</param>
1467     /// <param name="radiobutton">The radiobutton.</param>
1468     /// <param name="version">The version.</param>
1469     /// <param name="bestandsnaamArgument">The bestandsnaam argument.</param>
1470     /// <autogeneratedoc />
1471     /// TODO Edit XML Comment Template for Teststap
1472     public static void Teststap(IWebDriver driver, string machinestatic,
1473         string radiobutton, string version,
1474         string bestandsnaamArgument, bool opslaan, string saveWat = "")
1475     {
1476         var table = General.GetTestrun(bestandsnaamArgument);
1477         var actions = new Actions(driver);
1478         // var credits = GetCredentials(bestandsnaamArgument);
1479
1480         var inhoud = "";
1481         var vestandsnaam = "";
1482         var credits =
1483             GetCredentials(_project + @"\" + bestandsnaamArgument);
1484
1485         var machineNumber = Convert.ToString(getMachineCode());
1486
1487         for (var intx = 0; intx < table.Rows.Count; intx++)
1488         {
1489             var strikt = true;
1490             var savedAttribute =
1491                 table.Rows[intx]["save_attribute"].ToString().Trim();
1492             var drTitle = driver.Title;
1493             if (drTitle != null)
1494             {
1495                 vestandsnaam =
1496                     table.Rows[intx]["testcase"].ToString().Trim();
1497                 setClassName(table.Rows[intx]["testcase"].ToString().Trim());
1498                 setTestnr(table.Rows[intx]["testnr"].ToString().Trim());
1499
1500                 if (table.Rows[intx]["test_check_strict"].ToString().ToUpper() == "TRUE")
1501                     strikt = true;
1502                 else
1503                     strikt = false;

```

```

1506
1507         var v = (table != null
1508             ? table != null ? table.Rows[intx]["testurl"] : null
1509             : null) != string.Empty;
1510
1511         if (table.Rows[intx]["testurl"] != null &&
1512             table.Rows.Count > intx && v)
1513             if (table.Rows[intx]["testurl"].ToString().Trim()
1514                 .Length > 0
1515                 && table.Rows[intx]["testswitch"].ToString()
1516                     .ToUpper() == "YES")
1517             {
1518                 var handles = driver.WindowHandles;
1519                 foreach (var handle in handles)
1520                 {
1521                     var contains = driver.SwitchTo().Window
1522                         (handle)
1523                         .Url
1524                         .Contains(table.Rows[intx]["testurl"]
1525                             .ToString().Trim());
1526                 }
1527             }
1528         try
1529         {
1530             if (table.Rows[intx]["testaction"].ToString() ==
1531                 "wait")
1532             {
1533                 Thread.Sleep(Convert.ToInt32(
1534                     table.Rows[intx]["testtext"].ToString()
1535                         .Trim()) * 1000);
1536             }
1537             else
1538             {
1539                 var idString = table.Rows[intx]["testelement"]
1540                     .ToString().Trim();
1541                 var testname = table.Rows[intx]["testname"]
1542                     .ToString().Trim();
1543                 var testcase = table.Rows[intx]["testcase"]
1544                     .ToString().Trim();
1545                 var testaction = table.Rows[intx]["testaction"]
1546                     .ToString().Trim();
1547
1548                 if(idString != string.Empty)
1549                 {
1550                     Assert.IsTrue(
1551                         CheckBy(
1552                             driver,
1553                             table.Rows[intx]
1554                                 ["testelement"].ToString()
1555                                 .Trim(),
1556                             table.Rows[intx]
1557                                 ["testattribute"].ToString()
1558                                 .Trim(),
1559                             table.Rows[intx]["testnr"].ToString()
1560                                 .Trim(),

```

```
1558         radiobutton,
1559         table.Rows[intx]
        ["testtext_check"].ToString()
1560         .Trim(),
1561         table.Rows[intx]
        ["testaction"].ToString()
1562         .Trim(),
1563         table.Rows[intx]["testurl"].ToString
        ()
1564         .Trim(),
1565         table.Rows[intx]
        ["testswitch"].ToString()
1566         .Trim(),
1567         table.Rows[intx]["testdescription"]
1568         .ToString().Trim(),
1569         table.Rows[intx]
        ["testinverse"].ToString()
1570         .Trim(),
1571         table.Rows[intx]
        ["test_comment"].ToString()
1572         .Trim(),
1573         table.Rows[intx]["testname"].ToString
        ()
1574         .Trim(),
1575         table.Rows[intx]["testcase"].ToString
        ()
1576         .Trim(),
1577         version,
1578         0,
1579         machineNumber,
1580         table.Rows[intx]
        ["testlogicalobjectname"]
1581         .ToString().Trim(),
1582         credits.Url,
1583         table.Rows[intx]["testtext"].ToString
        ()
1584         .Trim(),
1585         bestandsnaamArgument,
1586         "",
1587         opslaan,
1588         strikt
1589     ));
1590 }
1591
1592
1593 IWebElement content = null;
1594 var tekst1 = string.Empty;
1595
1596 var testElement = table.Rows[intx]["testelement"]
1597     .ToString().Trim();
1598
1599
1600
1601
1602 var testAttribute =
1603     table.Rows[intx]["testattribute"].ToString()
```

```
1604         .Trim()
1605         .ToUpper();
1606     if (testElement != string.Empty)
1607     {
1608         switch (testAttribute)
1609         {
1610             case "ID":
1611                 content =
1612                     driver.FindElement(By.Id
1613 (testElement));
1614                 break;
1615             case "NAME":
1616                 content =
1617                     driver.FindElement(
1618                         By.Name(testElement));
1619                 break;
1620             case "CLASS":
1621                 content =
1622                     driver.FindElement(
1623                         By.ClassName(testElement));
1624                 break;
1625             case "XPATH":
1626                 content =
1627                     driver.FindElement(
1628                         By.XPath(testElement));
1629                 break;
1630             case "CSSSELECTOR":
1631                 content =
1632                     driver.FindElement(
1633                         By.CssSelector(testElement));
1634                 break;
1635         }
1636     }
1637     if (table.Rows[intx]["testelement"].ToString()
1638         .Trim().Length == 0
1639         || table.Rows[intx]["testelement"].ToString()
1640         .ToUpper().Trim() == "POPUP")
1641     {
1642         if (table.Rows[intx]["testtext"].ToString()
1643             .ToUpper() == "DISMISS")
1644             ExpectedConditions.AlertIsPresent()
1645                 .Invoke(driver).Dismiss();
1646         if (table.Rows[intx]["testtext"].ToString()
1647             .ToUpper() == "ACCEPT")
1648             ExpectedConditions.AlertIsPresent()
1649                 .Invoke(driver).Accept();
1650     }
1651
1652     if (table.Rows[intx]["testaction"].ToString() !=
1653         null)
1654         if (table.Rows[intx]["testaction"].ToString()
1655             .Trim().Length > 0
1656             || table.Rows[intx]
1657                 ["testaction"].ToString()
```

```

1658         .Trim().ToLower() != "empty")
1659     switch (table.Rows[intx]["testaction"]
1660         .ToString())
1661     {
1662         case "switch to url":
1663             driver.Navigate()
1664                 .GoToUrl(
1665                 table.Rows[intx]
1666                     .ToString());
1667
1668             inhoud +=
1669                 "driver.Navigate().GoToUrl('"
1670                 +
1671                 table.Rows[intx]["testurl"] +
1672                 "')\r\n";
1673             break;
1674         case "click":
1675             actions = new Actions(driver);
1676             actions.MoveToElement(content);
1677             actions.Perform();
1678             content.Click();
1679             break;
1680         case "value":
1681             if (table.Rows[intx]
1682                 ["testtext"] !=
1683                 null && table.Rows.Count >
1684                 intx
1685                 && (string)table.Rows[intx][
1686                     "testtext"]
1687                     != string.Empty)
1688             {
1689                 tekst1 = TekstVervanging(
1690                     table.Rows[intx]
1691                         .ToString().Trim());
1692                 actions = new Actions
1693                     (driver);
1694                 actions.MoveToElement
1695                     (content);
1696                 actions.Perform();
1697                 content.Click();
1698                 content.Clear();
1699                 content.SendKeys(tekst1);
1700             }
1701             break;
1702         case "sendkeys":
1703             if (table.Rows[intx]
1704                 ["testtext"] !=
1705                 null && table.Rows.Count >
1706                 intx
1707                 && (string)table.Rows[intx][
1708                     "testtext"]
1709                     != string.Empty)

```



```

1705         tekst1 = TekstVervanging(
1706             table.Rows[intx]
1707                 ["testtext"]
1708                 .ToString().Trim());
1709         if (table.Rows[intx][
1710             "test_password"] != null &&
1711             table.Rows.Count > intx
1712             && table.Rows[intx][
1713                 "test_password"].ToString
1714                 ()
1715                 != string.Empty)
1716         {
1717             tekst1 = TekstVervanging(
1718                 table.Rows[intx][
1719                     "test_password"]
1720                     .ToString().Trim());
1721             actions = new Actions(driver);
1722             actions.MoveToElement(content);
1723             actions.Perform();
1724             var tekst27 = tekst1;
1725             if (tekst1.ToUpper()
1726                 .Contains("{ENTER}"))
1727             {
1728                 var splitTekst =
1729                     tekst1.Split('{');
1730                 content.Click();
1731                 content.Clear();
1732                 content.SendKeys(splitTekst
1733                     [0] +
1734                     Keys.Enter);
1735             }
1736             else
1737             {
1738                 switch (tekst1.ToUpper())
1739                 {
1740                     case "TAB":
1741                         tekst1 = Keys.Tab;
1742                         break;
1743                     case "PGUP":
1744                         tekst1 = Keys.PageUp;
1745                         break;
1746                     case "PGDN":
1747                         tekst1 =
1748                             Keys.PageDown;
1749                         break;
1750                     case "ENTER":
1751                         tekst1 = Keys.Enter;
1752                         break;
1753                     case "ESC":
1754                         tekst1 = Keys.Escape;
1755                         break;
1756                 }
1757             }
1758         }

```

```

1756         content.Click();
1757         if (tekst27.ToUpper() !=
1758             "ENTER" &&
1759             tekst27.ToUpper() !=
1760             "TAB" &&
1761             tekst27.ToUpper() !=
1762             "PGUP" &&
1763             tekst27.ToUpper() !=
1764             "PGDN" &&
1765             tekst27.ToUpper() !=
1766                 "ESC")
1767             content.Clear();
1768             content.SendKeys(tekst1);
1769         }
1770
1771         break;
1772
1773         case "select":
1774             var tekst2 = TekstVervanging(
1775                 table.Rows[intx]["testtext"]
1776                     .ToString().Trim());
1777
1778             IWebElement education = null;
1779             SelectElement selectElement =
1780                 null;
1781
1782             switch (table.Rows[intx][
1783                 "testattribute"].ToString
1784                 ()
1785                 .Trim().ToUpper())
1786             {
1787                 case "ID":
1788                     // select the drop down
1789                     list
1790                     education =
1791                         driver.FindElement(
1792                             By.Id(table.Rows[
1793                                 intx][
1794                                     "testelement"]
1795                                     .ToString()
1796                                     .Trim()));
1797                     break;
1798                 case "NAME":
1799                     education =
1800                         driver.FindElement(
1801                             By.Name(
1802                                 table.Rows
1803                                     [intx]
1804                                     ["testelement"]
1805                                     .ToString
1806                                     ()
1807                                     .Trim
1808                                     ());

```

```

1803                                     break;
1804                                     case "CLASS":
1805                                         education =
1806                                             driver.FindElement(
1807                                                 By.ClassName(
1808                                                     table.Rows
1809                                                         [intx]
1810                                                         ["testelement"]
1811                                                         .ToString
1812                                                         ()
1813                                                         .Trim
1814                                                         ());
1815                                         break;
1816                                     case "XPATH":
1817                                         education =
1818                                             driver.FindElement(
1819                                                 By.XPath(
1820                                                     table.Rows
1821                                                         [intx]
1822                                                         ["testelement"]
1823                                                         .ToString
1824                                                         ()
1825                                                         .Trim
1826                                                         ());
1827                                         break;
1828                                     case "CSSSELECTOR":
1829                                         education =
1830                                             driver.FindElement(
1831                                                 By.CssSelector(
1832                                                     table.Rows
1833                                                         [intx]
1834                                                         ["testelement"]
1835                                                         .ToString
1836                                                         ()
1837                                                         .Trim
1838                                                         ());
1839                                         break;
1840                                     }
1841                                     //create select element object
1842                                     selectElement =
1843                                         new SelectElement(education);
1844                                     // select by text
1845                                     selectElement.SelectByText
1846                                         (tekst2);
1847                                     break;
1848                                     case "move_to":
1849                                         actions = new Actions(driver);
1850                                         actions.MoveToElement(content);
1851                                         actions.Perform();
1852                                         break;
1853                                     case "upload":

```

```

1846         tekst1 = TekstVervanging(
1847             table.Rows[intx]["testtext"]
1848             .ToString().Trim());
1849         var file1 =
1850             tekst1.Replace("\'", "\\'");
1851         content.Click();
1852         Thread.Sleep(5000);
1853         SendKeys.SendWait(file1);
1854         SendKeys.SendWait(@"{Enter}");
1855         break;
1856     case "checkbox":
1857         IWebElement checkbox1 = null;
1858
1859         checkbox1 = HaalElementop(driver,
1860             testElement, testAttribute);
1861
1862         var oldScript =
1863             checkbox1.GetAttribute(
1864                 "checked");
1865
1866         var tekst3 =
1867             table.Rows[intx]["testtext"]
1868             .ToString().Trim();
1869         var rc =
1870             (RemoteWebElement)checkbox1;
1871         var driver1 =
1872             (IJavaScriptExecutor)rc
1873             .WrappedDriver;
1874         var script1 = "";
1875         if (tekst3.ToUpper() == "TRUE" &&
1876             oldScript == null)
1877             script1 =
1878                 "arguments
1879 [0].setAttribute('checked', 'true');"
1880         if (tekst3.ToUpper() == "FALSE"
1881 &&
1882             oldScript != null)
1883             script1 =
1884                 "arguments
1885 [0].removeAttribute('checked');"
1886         driver1.ExecuteScript(script1,
1887 rc);
1888         break;
1889     case "switch to iFrame":
1890         var tekstFrame =
1891             table.Rows[intx]["testtext"]
1892             .ToString().Trim();
1893         var detailFrame = content;
1894         driver.SwitchTo()
1895             .Frame(detailFrame);
1896         break;
1897     case "switch to default":
1898         driver.SwitchTo().DefaultContent
1899 ();
1900         break;

```

```

1897         case "set_value":
1898             var query = string.Empty;
1899
1900             query =
1901                 "SELECT * FROM saved_values
1902
1903                 WHERE ";
1904
1905             query +=
1906                 "saved_values_testname = '" +
1907                 bestandsnaamArgument + "' AND
1908
1909                 ";
1910
1911             query += "saved_values_name = '"
1912
1913             +
1914                 table.Rows[intx]["testtext"]
1915                 .ToString().Trim() +
1916
1917                 "'";
1918
1919             var dt =
1920                 GenericDataRead.GetData
1921
1922             (query);
1923
1924             if (dt.Rows.Count > 0)
1925             {
1926                 // Indien variable voor
1927                 testscript bestaat, update deze uit tabel 'saved_values'
1928                 var save_id = dt.Rows[0][0]
1929                 .ToString();
1930                 query =
1931                     "UPDATE saved_values SET
1932
1933                     saved_values_value = '";
1934
1935                     if (savedAttribute ==
1936
1937                     string.Empty)
1938
1939                         query +=
1940
1941                         content.Text.Trim()
1942
1943                         .Replace("\r\n",
1944                             string.Empty)
1945
1946                         +
1947
1948                         "' ";
1949                     if (savedAttribute.ToUpper()
1950
1951                     == "HREF")
1952
1953                         query +=
1954
1955                         content.GetAttribute(
1956
1957                             "href").Trim()
1958
1959                             +
1960
1961                             "' ";
1962                     if (savedAttribute.ToUpper()
1963
1964                     == "ID")
1965
1966                         query +=
1967
1968                         content.GetAttribute(
1969
1970                             "id").Trim()
1971
1972                             +
1973
1974                             "' ";
1975                     if (savedAttribute.ToUpper()
1976
1977                     == "NAME")
1978
1979                         query +=
1980
1981                         content.GetAttribute(
1982
1983                             "name").Trim
1984
1985                         ()

```

```

1936                                     +
1937                                     "' ";
1938                                     query += "WHERE ";
1939                                     query += "saved_values_id = " +
1940                                     +
1941                                     save_id + ";";
1942                                     }
1943                                     else
1944                                     {
1945                                     // anders Voeg de gegevens
1946                                     toe aan de tabel 'saved_values'
1947                                     query =
1948                                     "INSERT INTO saved_values
1949                                     (saved_values_testname, "
1950                                     query +=
1951                                     "saved_values_name,
1952                                     saved_values_value, projectid) ";
1953                                     query += "VALUES(' " +
1954                                     bestandsnaamArgument +
1955                                     "' , "
1956                                     query += "' " +
1957                                     table.Rows[intx]
1958                                     ["testtext"]
1959                                     .ToString().Trim() +
1960                                     "' , '";
1961                                     if (savedAttribute ==
1962                                     string.Empty)
1963                                     query +=
1964                                     content.Text.Trim()
1965                                     .Replace("\r\n",
1966                                     string.Empty)
1967                                     +
1968                                     "' , ";
1969                                     if (savedAttribute.ToUpper()
1970                                     == "HREF")
1971                                     query +=
1972                                     content.GetAttribute(
1973                                     "href").Trim()
1974                                     +
1975                                     "' , ";
1976                                     if (savedAttribute.ToUpper()
1977                                     == "ID")
1978                                     query +=
1979                                     content.GetAttribute(
1980                                     "id").Trim()
1981                                     +
1982                                     "' , ";
1983                                     if (savedAttribute.ToUpper()
1984                                     == "NAME")
1985                                     query +=
1986                                     content.GetAttribute(
1987                                     "name").Trim
1988                                     ()
1989                                     +
1990                                     "' , ";

```

```

1977         query += getProjectID() +
1978             ");";
1979     }
1980     GenericDataRead.INUPDEL(query);
1981     break;
1982     case "get_value":
1983
1984         query =
1985             "SELECT saved_values_value,
1986             attribute FROM saved_values ";
1987         // query += "WHERE
1988         saved_values_testname = '' + bestandsnaamArgument + '' AND
1989         ";
1990         query += "WHERE ";
1991         query += "projectid = " +
1992             getProjectID() + " AND ";
1993         query += "saved_values_name = '"
1994             +
1995             table.Rows[intx]["testtext"]
1996             .ToString().Trim() +
1997             "';";
1998         var dtget =
1999             GenericDataRead.GetData
2000             (query);
2001
2002         if (dtget.Rows.Count > 0)
2003         {
2004             if (dtget.Rows[0][1]
2005                 .ToString() ==
2006                 string.Empty)
2007             {
2008                 actions =
2009                     new Actions(driver);
2010                 actions.MoveToElement(
2011                     content);
2012                 actions.Perform();
2013                 var tekstget =
2014                     dtget.Rows[0][0]
2015                     .ToString();
2016                 content.Click();
2017                 content.Clear();
2018                 content.SendKeys
2019                     (tekstget);
2020             }
2021             if (dtget.Rows[0][1]
2022                 .ToString().ToUpper() ==
2023                 "HREF")
2024             {
2025                 driver.Navigate()
2026                     .GoToUrl(
2027                         dtget.Rows[0][0]
2028                         .ToString());
2029             }
2030             if (dtget.Rows[0][1]

```

```

2023         .ToString().ToUpper()
2024         == "ID")
2025         {
2026             MessageBox.Show("Waarde
2027             ID: " + dtget.Rows[0][0].ToString());
2028             if (dtget.Rows[0][1]
2029                 .ToString().ToUpper()
2030                 == "NAME")
2031             {
2032                 MessageBox.Show("Waarde
2033                 Name: " + dtget.Rows[0][0].ToString());
2034             }
2035             break;
2036         case "scroll":
2037             var richting =
2038                 table.Rows[intx]["testtext"]
2039                 .ToString().Trim();
2040             var js =
2041                 (IJavaScriptExecutor)driver;
2042             if (richting.ToUpper() ==
2043                 "BOTTOM")
2044                 js.ExecuteScript(
2045                     "window.scrollTo(0,
2046                     document.body.scrollHeight)");
2047             if (richting.ToUpper() == "TOP")
2048                 js.ExecuteScript(
2049                     "window.scrollTo(0, 0)");
2050             if (richting.ToUpper() == "DOWN")
2051                 js.ExecuteScript(
2052                     "window.scrollTo(0,
2053                     1000)");
2054             if (richting.ToUpper() == "UP")
2055                 js.ExecuteScript(
2056                     "window.scrollTo(0,
2057                     1000)");
2058             break;
2059         case "cmd":
2060             if (intx != 0)
2061                 using (var cmd =
2062                     Process.Start(
2063                         table.Rows[intx][
2064                             "testtext"]
2065                             .ToString().Trim
2066                             ()))
2067             {
2068                 cmd.WaitForExit();
2069             }
2070             break;
2071         case "screenshot":
2072             var screenshotsPath =
2073                 GetCurrentDir(1);
2074             var pathG =

```



```

2070         screenshotsPath.Split('\\');
2071         var g = string.Empty;
2072
2073         for (var i = 0;
2074             i < pathG.Length - 2;
2075             i++) g += pathG[i] + "\\";
2076
2077         screenshotsPath = g;
2078         var name1 = applicationName.Trim
2079
2080         (
2081             );
2082
2083         var relativePath =
2084
2085         screenshotsPath +
2086
2087         "screenshots\\" + name1 + "\
2088
2089         \";
2090
2091         ScreenshotName =
2092         relativePath + name1 + "-" +
2093         string.Format(
2094             "{0:yyyyMMddHHmmssFFF}",
2095             DateTime.UtcNow)
2096         + "testnr_" + SetTestnr +
2097         ".png";
2098         General.LogMessage(
2099             "Screenshot name: " +
2100             ScreenshotName, 1);
2101         driver.TakeScreenshot()
2102             .SaveAsFile(ScreenshotName,
2103
2104             ScreenshotImageFormat.Png);
2105
2106         break;
2107
2108         case "logout":
2109             driver.Manage().Cookies
2110                 .DeleteAllCookies();
2111             break;
2112         }
2113
2114         if (testElement == string.Empty)
2115             General.LogMessage("", 1, testname,
2116                 Convert.ToInt32(testnr), testcase,
2117                 testaction, testAttribute);
2118         else
2119             General.LogMessage(
2120                 "Element " + testElement + "
2121
2122                 gevonden!",
2123
2124                 1, testname,
2125                 Convert.ToInt32(testnr), testcase,
2126                 testaction, testAttribute);
2127         }
2128
2129         /*var appendtext = GetCurrentDir(2) + "\\\" +
2130             vestandsnaam + ".js";
2131         var sw = File.CreateText(appendtext);
2132         sw.WriteLine(inhoud);
2133

```

```
2121         sw.Close(); */
2122     }
2123     catch (Exception exception)
2124     {
2125         General.LogMessage(exception.Message, 4,
2126             table.Rows[intx]["testname"].ToString().Trim(),
2127             Convert.ToInt32(table.Rows[intx]["testnr"]
2128                 .ToString().Trim()), string.Empty,
2129             machinestatic);
2130     }
2131 }
2132 else
2133 {
2134     MessageBox.Show("e.Message");
2135     break;
2136 }
2137 }
2138 }
2139
2140
2141 /// <summary>
2142 ///     Verifies the specified element name.
2143 /// </summary>
2144 /// <param name="elementName">Name of the element.</param>
2145 /// <returns><c>true</c> if XXXX, <c>false</c> otherwise.</returns>
2146 /// <autogeneratedoc />
2147 /// TODO Edit XML Comment Template for verify
2148 public static bool verify(string elementName)
2149 {
2150     try
2151     {
2152         return true;
2153     }
2154     catch
2155     {
2156         return false;
2157     }
2158 }
2159 }
2160 }
```