



CS 458
Software Verification & Validation
2019-2020 Spring

PROJECT #1

Berk Yıldız 21502040

1. Implementation of Stars Login Page

Stars Login Page is implemented by JavaScript and HTML. Project consists of two classes: *main.js* and *index.html*. *Index.html* creates the skeleton of the webpage and *main.js* brings interactivity into webpage.

```
JS main.js > ...
1
2 var accounts = [
3 > { ...
6   },
7   {
8     bilkent_ID: "7654321",
9     password: "something1"
10  },
11  {
12     bilkent_ID: "9362845",
13     password: "something3"
14  }
15 ]
16 function getInfo() {
17   var bilkent_ID = document.getElementById("bilkent_ID").value;
18   var password = document.getElementById("password").value;
19
20   if (bilkent_ID == "" && password == "") {
21     alert("Bilkent ID and password cannot be blank!")
22     return;
23   }
24   else if (bilkent_ID == "" || password == "") {
25     if (bilkent_ID == "") {
26       alert("Bilkent ID cannot be blank!")
27       return;
28     }
29     else if (password == "") {
30       alert("Password cannot be blank!")
31       return;
32     }
33   }
34   for (i = 0; i < accounts.length; i++) {
35     if (bilkent_ID == accounts[i].bilkent_ID && password == accounts[i].password) {
36       alert("You are logged in.")
37       return;
38     }
39     else {
40       alert("Bilkent ID or password is wrong!")
41       return;
42     }
43   }
44 }
```

Figure 1: JavaScript code


```

<> index.html > ...
1
2  <!DOCTYPE html>
3  <html>
4
5  <head>
6      <title>Bilkent Accounts :: Login </title>
7      <meta charset="utf-8">
8      <link rel="stylesheet" href="main.css">
9  </head>
10
11 <body>
12     <section>
13         <div class="logo">
14             <a href="file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogin/index.html">
15                 
16             </a>
17         </div>
18         <div class="logo2">
19             
20         </div>
21     </section>
22     <form id="form1">
23         <label for="bilkent_ID">Bilkent ID:</label>
24         <input id="bilkent_ID">
25         <br><br>
26         <label for="password">Password:</label>
27         <input id="password" type="password"><br><br>
28         <button id="loginButton" type="button" onclick="getInfo()">Login</button>
29     </form>
30     <script src="main.js"></script>
31 </body>
32
33 </html>

```

Figure 2: HTML code

Bilkent University Online Services



Bilkent University

Secure Login Gateway

Login : **SRS** - Student Academic Information Registration System

Bilkent ID:

Password:

Figure 3: Appearance of Login Page

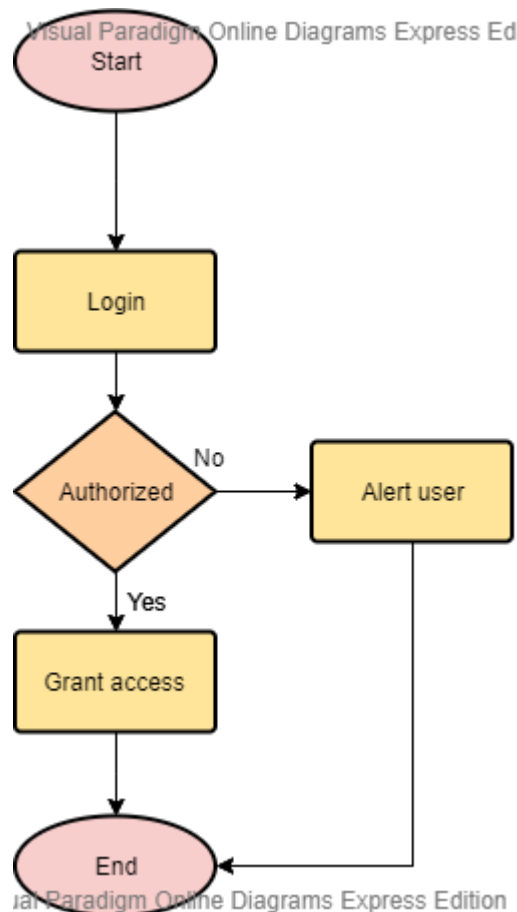


Figure 4: Login Process

2. Capabilities of Selenium

- The existence of requested components on a webpage can be verified by Selenium.
- Giving correct error messages can be verified by Selenium.
- Test automation codes can be written in many languages like Java, Python, JavaScript etc.
- Test cases for valid and invalid inputs for the elements of a webpage can be verified by Selenium.
- There are several ways for selecting components on a webpage by using cssSelector, name, id, xpath.
- It provides user-friendly IDE which is a Chrome and Firefox extension to speed up developing test cases.
- Setting the environment for writing test automation code by Selenium is very easy.
- It supports various operating systems and browsers.
- It supports several frameworks like NUnit, JUnit, TestNG etc.
- It provides huge number of methods to examine every possible scenario on a webpage.

3. Test Cases & Automation Codes

Enter valid Bilkent ID & password and click login. Verify if user can login to system with existing Bilkent ID and password.

```
driverChrome.navigate().to("file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogin/index.html");
driverChrome.findElement(By.cssSelector("#bilkent_ID")).sendKeys("1234567");
driverChrome.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
driverChrome.findElement(By.cssSelector("#password")).sendKeys("something");
driverChrome.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
driverChrome.findElement(By.cssSelector("#loginButton")).click();
message = driverChrome.switchTo().alert().getText();
if(message.equals("You are logged in.)) System.out.println("Test for valid login is successful");
else System.out.println("Test for valid login fails");
```

Enter invalid Bilkent ID & password combination and click login. Verify if user cannot login to system and “Bilkent ID or password is wrong!” message shows up.

```
driverChrome.navigate().to("file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogin/index.html");
driverChrome.findElement(By.cssSelector("#bilkent_ID")).sendKeys("1234567");
driverChrome.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
driverChrome.findElement(By.cssSelector("#password")).sendKeys("something1");
driverChrome.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
driverChrome.findElement(By.cssSelector("#loginButton")).click();
message = driverChrome.switchTo().alert().getText();
if(message.equals("Bilkent ID or password is wrong!)) System.out.println("Test for invalid login is successful");
else System.out.println("Test for invalid login fails");
```

Leave Bilkent ID & password blank and click login. Verify if “Bilkent ID and password cannot be blank!” message shows up.

```
driverChrome.navigate().to("file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogin/index.html");
driverChrome.findElement(By.cssSelector("#loginButton")).click();
message = driverChrome.switchTo().alert().getText();
if(message.equals("Bilkent ID and password cannot be blank!))
System.out.println("Test for blank BilkentID & password is successful");
else System.out.println("Test for blank BilkentID & password fails");
```

Enter password & leave Bilkent ID blank and click login. Verify if “Bilkent ID cannot be blank!” message shows up.

```
driverChrome.navigate().to("file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogin/index.html");
driverChrome.findElement(By.cssSelector("#password")).sendKeys("something");
```

```

driverChrome.manage().timeouts().implicitlyWait(3, TimeUnit.SECONDS);
driverChrome.findElement(By.cssSelector("#loginButton")).click();
message = driverChrome.switchTo().alert().getText();
if(message.equals("Bilkent ID cannot be blank!")) System.out.println("Test for
blank BilkentID is successful");
else System.out.println("Test for blank BilkentID fails");

```

Enter Bilkent ID & leave password blank and click login. Verify if “Password cannot be blank!” message shows up.

```

driverChrome.navigate().to("file:///C:/Users/BERKYILDIZ/Desktop/cs458/starsLogi
n/index.html");
driverChrome.findElement(By.cssSelector("#bilkent_ID")).sendKeys("1234567");
driverChrome.findElement(By.cssSelector("#loginButton")).click();
message = driverChrome.switchTo().alert().getText();
if(message.equals("Password cannot be blank!")) System.out.println("Test for
blank password is successful");
else System.out.println("Test for blank password fails");

```

4. Evaluation of Automation Experience

Before I started the project, I had no idea about implementation of automation, even though I knew about it as a concept. My mind was like a blank page for the coding side. But after I started to follow documentation and tutorials of Selenium, the idea of automation aroused curiosity. After I started to practice automation, I realised how can it operates the testing process more accurate and faster and understood having knowledge of this topic can be really useful for my self development. In general, I can say that I had a very nice experience for automation and I am very pleased to have an basic idea about automation.

5. Contribution of Automation to SDLC

A software would require to be tested with possible scenarios for several times to ensure quality and automation provides it in an acceptable time without a need to human. It increases the speed of testing process by presenting faster feedback and provides wider test coverage. It speeds up the process to detect defects earlier. Earlier detection of defects would help to improve both velocity and quality because early detection is important for ensuring correct functionality and earlier solved problems can pull forward the release date of the product. Also testing itself takes lots of time in software development lifecycle, so automation directly increases the efficiency by reducing the time spent on testing.