Unit testing

As you can see from the table bellows, here are some unit testing from the website, which allows to student to make a booking for seats or a group setting.

|  |  |  |  |
| --- | --- | --- | --- |
| Register | Log\_In | Send\_email | Log\_out |
| Allows the student to register into the database | Allows the student to log into the system, using the username and password | Used to send email to recover password | Allows the student to log out the system |
| Create\_student()  Validate\_name(string)  Validate\_email(string)  Validate\_password(string)  Validate\_username(string) | Check\_susername(string)  Check\_password(string) | recover\_psw(string) | Log\_out() |

After identifying the teste unit in the system, I tested all the methods and classes, by isolating and test the states using behavioural model and tested every possible stated transition sequences, which will provide all the coverage of all the feature of the class, also testing the set value of all attributes. This method will be called only when the user aka the student will enter the right data in the registration form and clicked on the submit button

Register method, takes different string parameters “name”,”password”,”email”,”username”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Full Name | Email Address | Username | Password | Expected Result | Actual Result | Pass/Fail |
| Lila Beck | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| lila beck | lila@gmail. | lila94 | 123456789 | Error, invalid email | Pop-up box, invalid email | Pass |
| lila | lila@gmail.com | lila94 | 1234 | Error, wrong password length | Pop-up box, wrong password length | Pass |
|  |  |  |  | Error, empty form | Pop-up box, empty form | Pass |
| LilaBeck1 | Lilagmail.com | lila94 | 123456789 | Error, wrong name and email value | Pop-up box, wrong name and email value | Pass |
| lila1 |  | lila94 | 123456789 | Error, wrong name value and missing email | Pop-up, wrong name value and missing email | Pass |

Validate\_name(string)

Validate\_name method, takes different string parameters “name”. Validates the name field before sending the data to the database

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Full Name | Email Address | Username | Password | Expected Result | Actual Result | Pass/Fail |
| Lila Beck | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| lila beck | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| lila | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
|  | lila@gmail.com | lila94 | 123456789 | Error, empty name | Pop-up box, empty name field | Pass |
| LilaBeck1 | lila@gmail.com | lila94 | 123456789 | Error, wrong name value | Pop-up box, wrong name value | Pass |
| lila@1 | lila@gmail.com | lila94 | 123456789 | Error, wrong name value | Pop-up, wrong name value | Pass |

Validate\_email(string)

Validate\_email method, takes different string parameters ”email”. Validates the email field before sending the data to the database

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Full Name | Email Address | Username | Password | Expected Result | Actual Result | Pass/Fail |
| Lila Beck | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| Lila Beck | lila@gmail. | lila94 | 123456789 | Error, incomplete email | Pop-up box, invalid email | Pass |
| Lila Beck | lilagmail.com | lila94 | 123456789 | Error, missing ‘@’ | Pop-up box, invalid email | Pass |
| Lila Beck | lila94@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| Lila Beck | @gmail.com | lila94 | 123456789 | Error, incomplete email | Pop-up box, wrong email format | Pass |
| Lila Beck | 123@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |

Validate\_password(string)

Validate\_password method, takes different string parameters ”password”. Validates the password field before sending the data to the database

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Full Name | Email Address | Username | Password | Expected Result | Actual Result | Pass  /Fail |
| Lila Beck | lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| Lila Beck | lila@gmail.com | lila94 | Password1 | Added into database | Added into database | Pass |
| Lila Beck | lila@gmail.com | lila94 | password123 | Added into database’ | Added into database | Pass |
| Lila Beck | lila@gmail.com | lila94 | password@123 | Added into database’ | Added into database | Pass |
| Lila Beck | lila@gmail.com | lila94 | 12345 | Error, not long enough | Pop-up box, invalid password | Pass |
| Lila Beck | lila@gmail.com | lila94 |  | Error, empty | Pop-up box, invalid password | Pass |

Validate\_username(string)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Email Address | Username | Password | Expected Result | Actual Result | Pass  /Fail |
| lila@gmail.com | lila94 | 123456789 | Added into database | Added into database | Pass |
| lila@gmail.com | lila | 123456789 | Added into database | Added into database | Pass |
| lila@gmail.com |  | 123456789 | Error, missing username | Pop-up box, missing username | Pass |
| lila@gmail.com | @lila | 123456789 | Added into database’ | Added into database | Pass |

Validate\_username method, takes different string parameters ”username” Validates the username field before sending the data to the database

Check\_username() Validates the name field matches the data in the database, Check\_username method, takes different string parameters ”c\_username”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Username | Password | Expected Result | Actual Result | Pass  /Fail |
| lila94 | 123456789 | Logged into system | Logged into system | Pass |
| lila | 123456789 | Logged into system | Logged into system | Pass |
|  | 123456789 | Error, missing username | Pop-up box, missing username | Pass |
| @lila | 123456789 | Logged into system | Logged into system | Pass |
| Lila86 | 123456789 | Error, wrong username | Error, wrong username | Pass |

Check\_password(string)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Username | Password | Expected Result | Actual Result | Pass  /Fail |
| lila94 | 123456789 | Logged into system | Logged into system | Pass |
| lila | 012345 | Error, wrong password | Pop-up box, wrong password | Fail |
| Lila86 |  | Error, missing password | Pop-up box, missing password | Fail |
| @lila | password | Error, wrong password | Pop-up box, wrong password | Fail |

Check\_password method, takes different string parameters ,”c\_password”, uses HTML code to previously validate the password and additional validation are made

Recover\_psw(string)

Recover\_psw method, takes different string parameters,”r\_email”. Will take the data from the database, the email provided and send an email which will allow the user to reset the password.

|  |  |  |  |
| --- | --- | --- | --- |
| Email | Expected Result | Actual Result | Pass  /Fail |
| lila@gmail.com | Email sent to provided email address | Logged into system | Pass |
| Lila94@gmail.com | Error, email does not exist on system | Logged into system | Pass |
|  | Error, empty email field | Pop-up box, missing username | Fail |

Log\_out(), the user should be able to log out, and when clicking the back button, the log in page be displayed

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Expected Result | Actual Result | Pass  /Fail |
| Click on log-out button | Logged out from the system | Logged into system | Fail |

Components Testing

I have tested the interaction between the website and the database, more precisely when the users enter the data into the form fields and clicks on register, where the data will travel to through the network to reach the database, to then can retrieve the send data to be displayed.

Testing Process

The whole testing process on the website, have been made by one member of the team, when testing different area of the system, I choose which was the most important for the users and when under different types of testing to make sure the system was error and bug free, to ensure high level of user satisfaction. When testing the different methods in the unit testing I made sure to enter the wrong type of value or format to make sure that if the user made the same mistake a warning or alert box will come up. When testing, I have considered which one was useful, for the unit testing, white box testing was used to test the internal (the code structure, code modules, source code, etc.), design and implementation of the methods within the website, as for the system testing, I have used the black box testing(test the design specification of the system), for system testing to test the interface and if the system is working as it should. Even if some of the testing have fail, I will be working on the code more in depth to make sure that all testes pass. No tools where used in this occasion when testing because the process was more easier and quicker to do, without compromising in the quality of the test, and simulating the errors a user can make while operating on the system.

Different issues have been encountered when testing, as it was a long and tedious job to be done by one person, but I made sure that the quality of the result have not been compromised, another issue encountered was correcting the failed testes to pass, as more in depth knowledge in the JavaScript is required. Another disadvantage was for the performance testing couldn’t be carried out properly.

As for the testing results, there were some failed tested within the system, as I am working to make sure that the system has no error, most of the tests were made in order to display an error message to the user, example of it when the user leaves am empty field, the system will prompt the user of the committed mistake, testing the components to make sure that they interacted properly was one of the most challenged testing I have personally named, as for the system testing, I have asked different types of user, some have technical background and some didn’t have any It background, which gave me a different type of feedback where I can make my changes and improve from it and noticed errors that “I” the developer completely overlooked and did notice. As for the performance testing, the results could have been comprised because of the server which did not perfume properly at the time, when testing, which is why most user have encounter slow and long loading time and the system crashed when just a small amount of user have logged into the system, or registering at the same time.