Lab 10

Software Engineering & Testing

Presented by:

Illia Stefanovskyi(B00165280), Jayson Baldemor(B00161560), Stella Bakac(B00159322)

Unit testing

Currently we have most of the unit testing done, zip file with the code is added to the submission. Unit tests showcase inheritance and aggregation examples. During the test we found a minor issue that user file wasn't required in sub classes, it was fixed and now everything works fine.

Data about manager retreived from employee

2

John1

0877837843

manager@gmail.com

Employee data

0

Duglas 1

GEO

9993337778

2

employee1@gmail.com

Equivalence Partition Testing

For this testing I created some if – else logic statements to test how the filters would work for the cats, based on gender and age. So, for example, if user picks "male" and the age is more than 1, it will show male cats. If it's "female" and the age is over 2, it shows female cats. I also added a condition where if user picks "any" gender and the age is above 0, then it just shows all the cats. Then I added when only gender is selected – it still works and shows all cats. Lastly, if the inputs are incorrect or it doesn't match, the system will not show any cats.

```
Equivalence Partition Testing:

if (gender == male && age > 1)
    show male cat;

else if (gender == female && age > 2)
    show female cat;

else if (gender == any && age > 0)
    show all cats;

else if (gender == male || gender == female)
    show all cats;

else if (age > 0 && age < 30)
    show all cats;

else //if the values are not correctly
    don't show the cats
```

Validation

```
<script>
const loginForm = document.querySelector('.left');
loginForm.addEventListener('submit', function(e) {
    const email = loginForm.querySelector('input[name="email"]').value.trim();
    const password = loginForm.querySelector('input[name="password"]').value.trim();

    if (!email || !password) {
        alert("Email and password are required!");
        e.preventDefault();
    }
});

//TODO check for password length, do it later to simplify development and testing

const signupForm = document.querySelector('.right');
signupForm.addEventListener('submit', function(e) {
    const email = signupForm.querySelector('input[name="email"]').value.trim();
    const password = signupForm.querySelector('input[name="password"]').value.trim();
    const confirmPassword = signupForm.querySelector('input[name="passwordConf"]').value.trim();
    const phone = signupForm.querySelector('input[name="phonenum"]').value.trim();

    if (password !== confirmPassword) {
        alert("Passwords do not match!");
        e.preventDefault();
    }

    if (!/^[0-9]{7,10}$/.test(phone)) {
        alert("Phone number should be digits only (7-10 numbers)!");
        e.preventDefault();
    }
    });
</script>
```

Validation is seen throughout the entire website, Inside these <scripts> is the JavaScript code specifically for validation in certain areas and functions. In this scenario, under the comments. It checks and validates if the password matches in the login.php page, there are also certain input filters such as when inputting a phone number, only digits from the NUM key are only inputted rather than other non-digit inputs such as symbols and letters.

```
    <script>
        document.querySelector("form").addEventListener("submit", function (event) {
            var minAge = document.querySelector("#minAge").value.trim();
            var maxAge = document.querySelector("#maxAge").value.trim();
            if (minAge > maxAge || minAge < 0 || maxAge > 30 || minAge == "" || minAge == "") {
                event.preventDefault();
                alert("Enter valid range between 0 and 30!");
            }
        });
    </script>
</div>

<button class="addFiltersButton buttonLink">Add filters</button>

<script>
    document.querySelector(".addFiltersButton").addEventListener("click", function () {
        const container = document.querySelector(".filtersContainer");
        if(container.style.display === "flex"){
            container.style.display = "none";
        }else{
            container.style.display = "flex";
        }
    });
</script>
```

Here is another example of Validation in the project/website

This is inside the index.php, This function works as a filter, a filter is for the cats ages ranging between 0 to 30. The validation in this is to make sure they input a proper value between 0 – 30.

This is the message that will pop up if the user inputs a invalid digit:

“event.preventDefault(); alert (“Enter valid range between 0 and 30!”); “