

SOFTWARE VERIFICATION, VALIDATION AND TESTING

TESTING DOCUMENTATION

REDDIT

Prepared by:

Amira Abdo

Muhamed Hamzić

Proposed to:

Samed Jukić, Assist. Prof. Dr.

Aldin Kovačević, Teaching Assistant

1. Introduction	3
1.1. About the project	3
1.2. Project Functionalities and Screenshots	4
2. Test Plan	6
2.1. Scope	6
2.2. Testing Environment and Tools	6
3. Test Execution	7
3.0 Reddit profile initializer	7
3.1 Reddit Avatar	9
3.2. Test general UI Features	14
3.3. Posts	22
3.4. Social links	26
3.5. Miscellaneous	29
3.5.1. Testing of Emoji Send	29
3.5.2 Test Join Subreddit	32
3.5.3. Test Leave Subreddit	33
3.5.4 Test Login User Flow	35
3.5.5 Test Message Delete	37

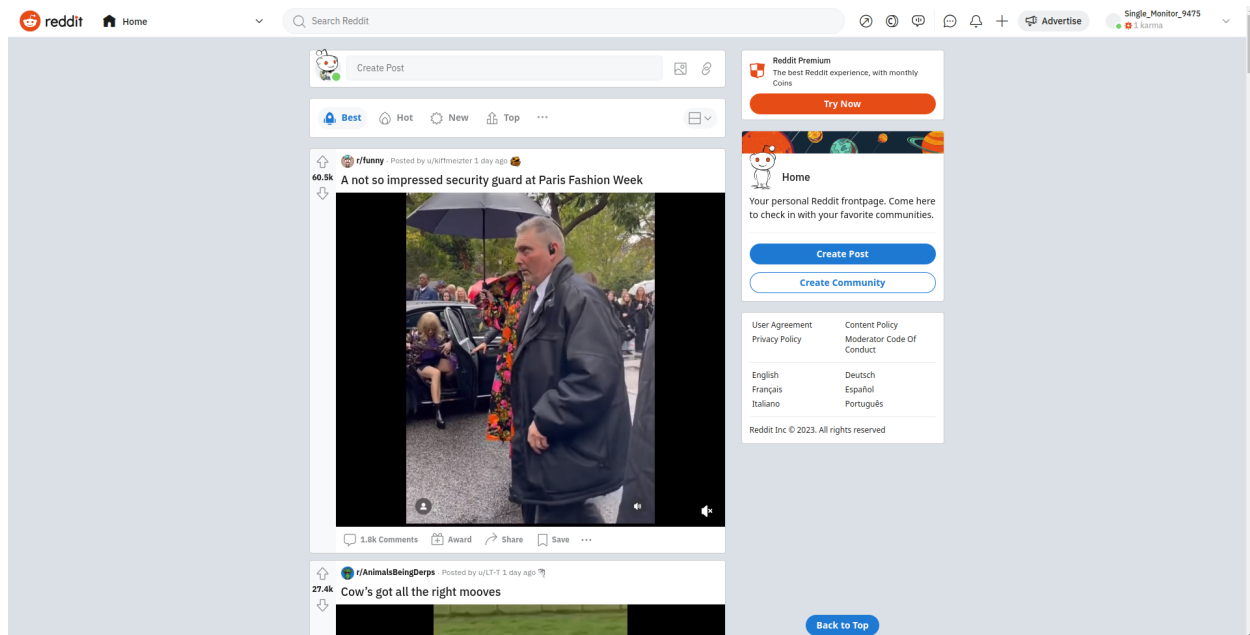
1. Introduction

1.1. About the project

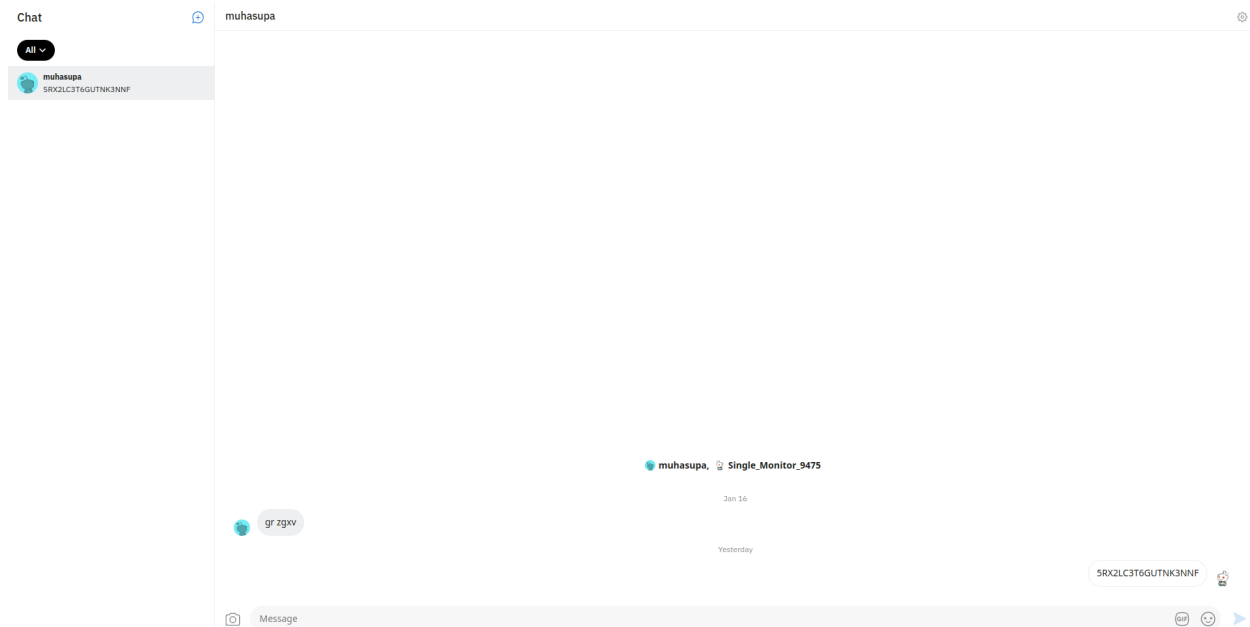
Reddit is a social news and discussion website where users can post and comment on content. It is organized into various topic-specific communities called "subreddits." Each subreddit has its own rules and moderators, and users can subscribe to as many subreddits as they want. Reddit also has a feature called "upvoting" and "downvoting," which allows users to signal their approval or disapproval of a post or comment. Reddit is known for its open and diverse community, as well as its role in hosting and promoting discussions on a wide range of topics.

1.2. Project Functionalities and Screenshots

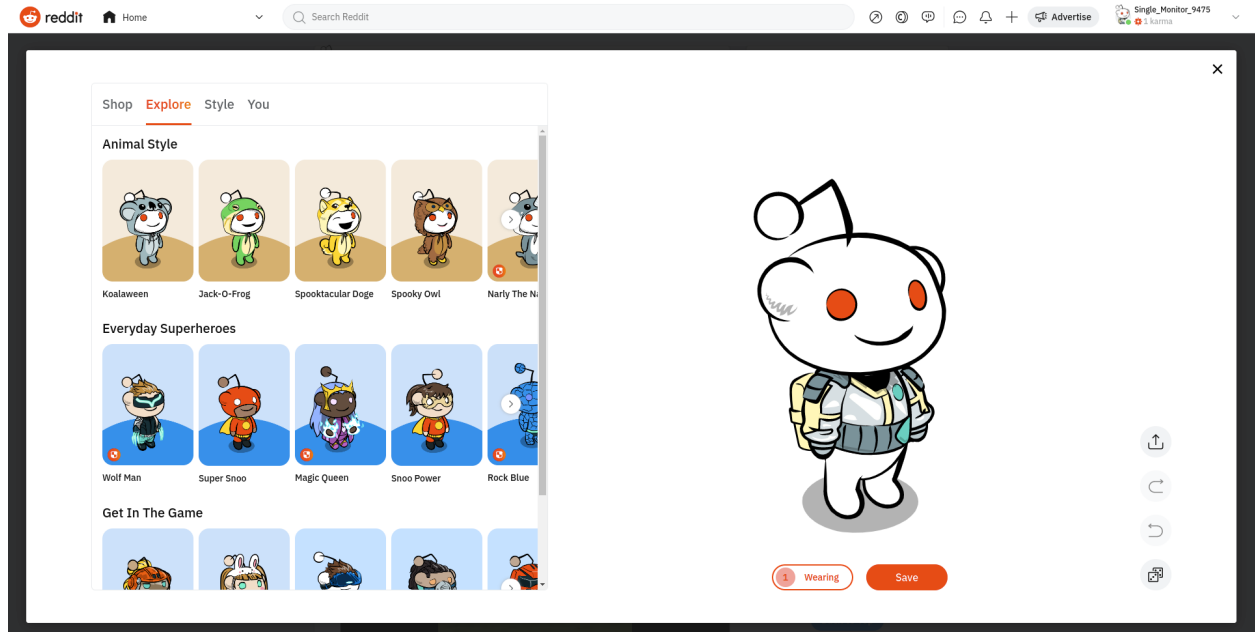
The main purpose and hence the functionalities of this project is to test various functions of Reddit.



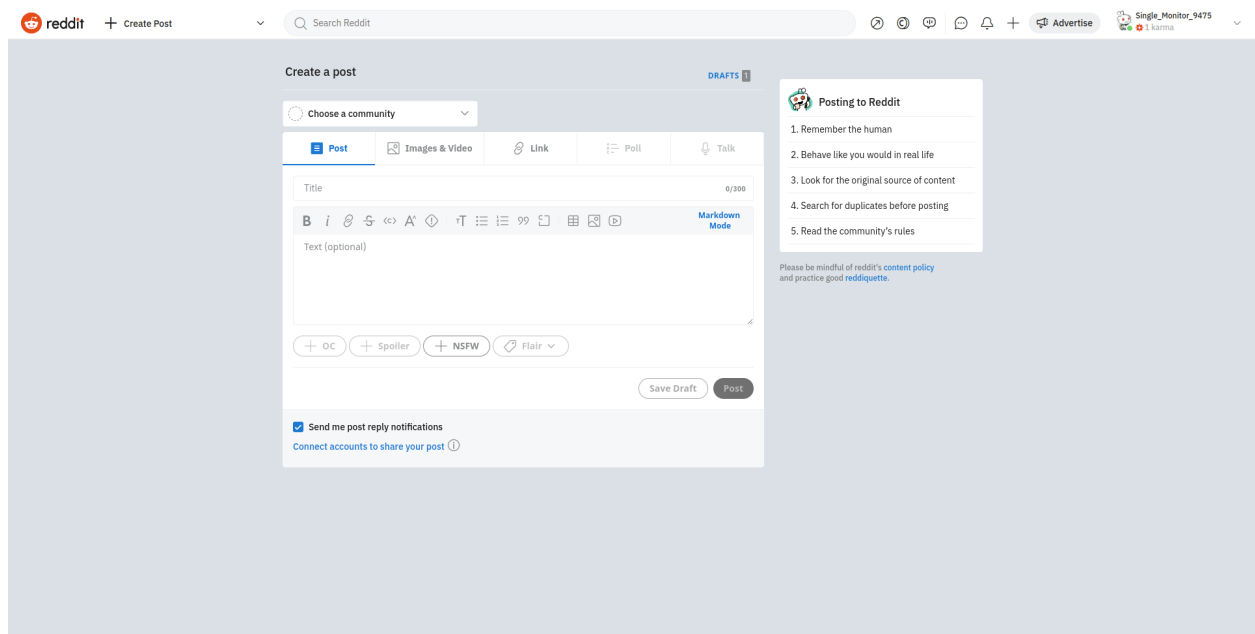
The homescreen where trending posts are found.



The chat section where the user can communicate with other users



The user can style their avatar.



The user can also create posts and save them to drafts or post them.

2. Test Plan

2.1. Scope

Since Reddit is a complex website with numerous features it would be nearly impossible to test all features properly in respect to the assigned ECTS points for the course.

Hence we have decided to focus on a number of “feature batches”. We define a feature batch as a closely related functionalities which enable the execution of a user flow.

Our scope for testing consists of:

- Reddit Avatar functionality, where we will test the creation, update and modification of it.
- Common user flows such as log in, log out.
- User communication functionality through chat where we will test gifs, emotes deletion and sending of messages
- Posting of comments and posts as well as post draft functionality and user reporting flow
- Miscellaneous test considering the UI

2.2. Testing Environment and Tools

For the project we are using the Java programming language. Additionally, we are using Maven version 4.0.0. as our project bootstrapper and dependency manager paired with junit version 5.7.1 in addition to Selenium 4.6.0.

All test have been performed on Google Chrome running on Fedora 36 and Windows

Chrome driver version: 109.0.5414.74 (Fedora 36)

Chrome driver version: 107.0.5304.62

3. Test Execution

3.0 Reddit profile initializer

Test Name: Reddit profile initializer				
Description: Initializes a chrome profile in order to perform other tests with ease				
Pre-condition(s): You must have a reddit account and know its credentials				
Test Steps: 1. Start the test	Test Data:	Expected Result: The user will use the opened Chrome tab to login into Reddit.	Actual Result: Depends on the user.	Status: TBA
Notes: This test is only intended for easier creation of ChromeProfiles				

```
class testChromeProfile {  
  
    /**  
     * This test can be used as a setup in order to create a Chrome Profile,  
     * it eliminates the need of constant log in  
     *  
     */  
    private static WebDriver webDriver;  
    private static String baseUrl;  
  
    @BeforeAll  
    static void setUp() {  
        System.setProperty("webdriver.chrome.driver", "/home/musaka/Desktop/Semester
```

```
5/svvt/chromedriver");
```

```
ChromeOptions options = new ChromeOptions();  
options.addArguments("--no-sandbox",  
                    "--disable-notifications",  
                    "--disable-extensions");  
String path = "/home/musaka/Desktop/RedditProfile";  
options.addArguments("--user-data-dir="+path);
```

```
options.addArguments("--start-maximized");
```

```
webDriver = new ChromeDriver(options);
```

```
baseUrl = "https://www.reddit.com/";
```

```
}
```

```
@AfterAll
```

```
static void tearDownAfterClass() throws Exception {  
    webDriver.quit();  
}
```

```
@Test
```

```
void testCommentPersistance() throws InterruptedException {  
    webDriver.get(baseUrl);  
    webDriver.manage().window().maximize();
```

```
    Thread.sleep(300000);
```

```
}
```

```
}
```


3.1 Reddit Avatar

Test Name: Test Avatar Body Color				
Description: The test performs a check of every preset available body color for the avatar				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Navigate to user profile 2. Open create avatar panel 3. Open you section 4. Open body section 5. Get all body options 6. Check every body option	Test Data:	Expected Result: The test goes through all body options and every body change is reflected in the avatar preview screen.	Actual Result: The actual result was identical to the expected result.	PASS
Notes: The test does not test custom colors.				

@Test

```
void testIfAvatarChangesColor() throws InterruptedException {  
    webDriver.get(baseUrl);  
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
```

```
    //open create avatar  
    wait.until(  

```

```
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/  
div[2]/div/div/div/div[2]/div[3]/div[2]/div/div[1]/div/button"))  
    ).click();
```

```
    //open you section
```

```
    wait.until(  

```

```

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/nav/div/div[4]"))
        ).click();
    //open body section
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/div/div/div[2]/section/div/div[1]"))
        ).click();
    // check body change color
    // get all color body options
    WebElement holder = wait.until(

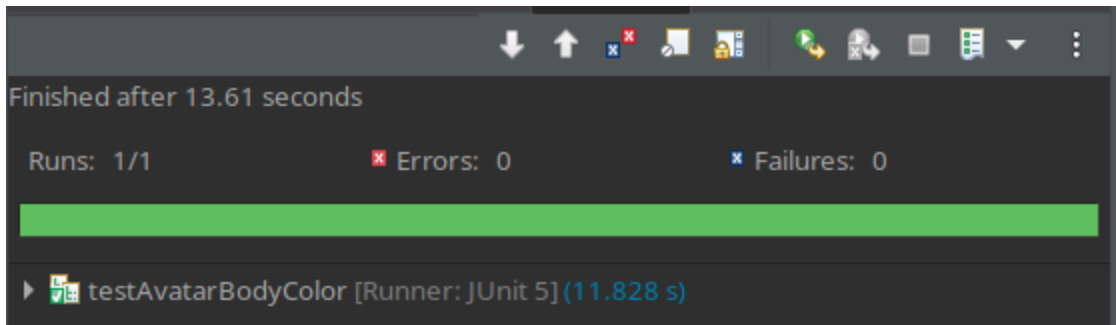
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/div/div/div[2]/section/div[2]/div[2]/div/div[1]/div/div"))
        );
    List<WebElement> colorOptions = holder.findElements(By.xpath("./child::*"));

    for(int i = 0; i < colorOptions.size()-1;i++) {
        colorOptions.get(i).click();
        System.out.print(colorOptions.get(i).getCssValue("background-color"));
        Thread.sleep(1000);
        WebElement body = webDriver.findElement(By.xpath("/html/body"));
        String style = body.getAttribute("style");
        int startIndex = style.indexOf("--color-body:");
        String color = style.substring(startIndex + 13 , startIndex + 20);

        int r = Integer.valueOf(color.substring(1,3),16);
        int g = Integer.valueOf(color.substring(3,5),16);
        int b = Integer.valueOf(color.substring(5,7),16);

        System.out.println(" - " + "rgba(" + r + ", " + g + ", " + b + ", 1)");
        assertEquals(colorOptions.get(i).getCssValue("background-color"),"rgba("
+ r + ", " + g + ", " + b + ", 1)" );
    }
} }

```



Test Name: Garment Change				
Description: It tests if Garment changes are reflected in the Reddit Avatar preview				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Open create avatar 2. Open style section 3. Open face section 4. Get the item holder 5. Get all face options 6. Check if they match	Test Data:	Expected Result: The avatar changes are properly reflected in the avatar preview screen	Actual Result: The actual result matches the expected result	PASS
Notes: The test tests only certain garments				

```

@Test
void testGarments() throws InterruptedException {
    webDriver.get(baseUrl);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

    //open create avatar
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[2]/div/div[1]/div/button"))
        ).click();
    //open style section

    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/nav/div/div[3]"))
        ).click();

    // open face section
    //
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/div/div/div[2]/div[4]"))
        ).click();
    // get the item holder
    WebElement itemHolder = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[1]/div/div/div[3]/div[2]/div/div[2]"))
        );
    // get all face options
    List<WebElement> faceOptions = itemHolder.findElements(By.xpath("./child::*"));
    List<WebElement> actualCurrentState = new ArrayList<WebElement>();
    for( WebElement faceOption : faceOptions) {
        //System.out.println(faceOption.getAttribute("class"));
        WebElement child = faceOption.findElement(By.xpath("./child::*"));
        WebElement child2 = child.findElement(By.xpath("./child::*"));
    }
}

```

```

//System.out.println(child2.getAttribute("data-testid"));

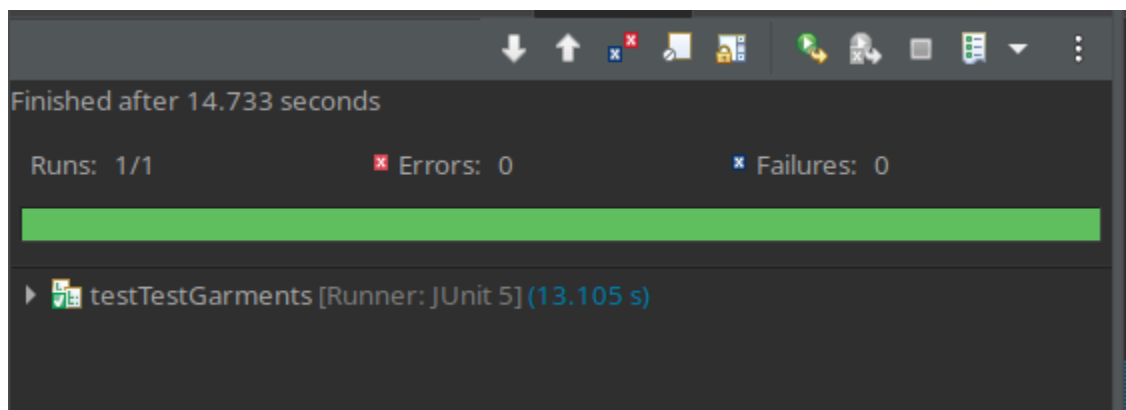
List<WebElement> children = child2.findElements(By.xpath("./child::*"));

//System.out.println(children.get(3).getAttribute("data-testid"));
actualCurrentState.add(children.get(3));

//System.out.println("--- End of children ---");
}

for( int i = 0;i < 10;i++) {
    faceOptions.get(i).click();
    String s = actualCurrentState.get(i).getAttribute("data-testid");
    String s1 =
webdriver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div/div[1]/div[1]/div[2]/div[2]/div/div[4]")).getAttribute("data-testid");
    assertEquals(s,s1);
    Thread.sleep(500);
}
}

```



3.2. Test general UI Features

Test Name: Display language change				
Description: Checks if UI properly responds to language change				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Navigate to settings 2. Wait till selects loads 3. Extract all available languages 4. Check all languages	Test Data: We expect for a certain element to change its value upon language change so we listen on 8 different ways	Expected Result: The listened on element will change its language based on select changes	Actual Result: The actual result is the same as the expected result.	PASS
Notes: The test does not check all elements for a change.				

```
@Test
void testDisplayLanuangeChange() throws InterruptedException {
    webDriver.get(baseUrl);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

    wait.until(
        ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div[4]/div[3]/span/select"))
    );

    Select language = new
    Select(webDriver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div[4]/div[3]/span/select")));

    List<WebElement> allLanguages = language.getOptions();
```

```

List<String> visibleTextLanguages = new ArrayList<String>();
for( int i = 0;i < allLanguages.size();i++) {
    visibleTextLanguages.add(allLanguages.get(i).getText());
}

for( int i = 0;i < allLanguages.size();i++) {
    System.out.println("Testing: " + visibleTextLanguages.get(i));
    language = new
Select(webDriver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div[4]/div[3]/span/select")));

    language.selectByVisibleText(visibleTextLanguages.get(i));
    wait.until(

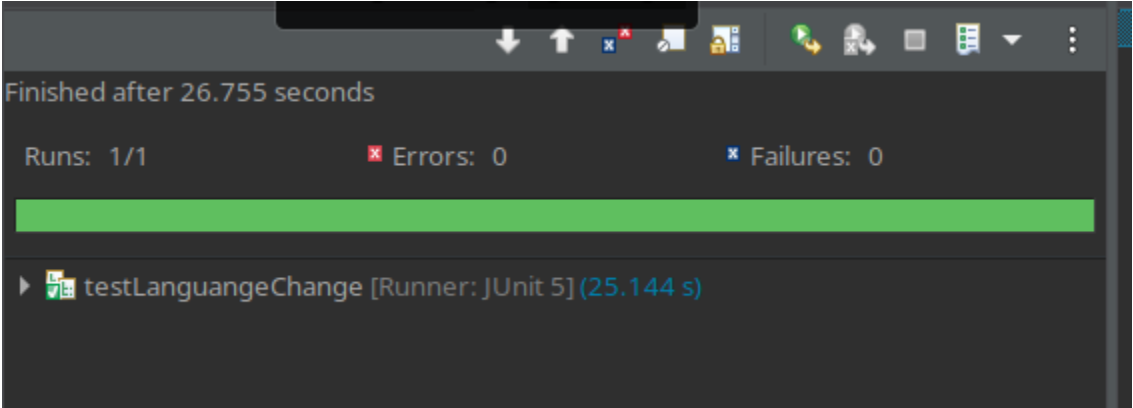
ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div[2]/div[1]/h2"))
    );
    Thread.sleep(300);

    String accSettingsText =
webDriver.findElement(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div[2]/div[1]/h2")).getText();

    if(visibleTextLanguages.get(i).equals("Deutsch")) {
        assertEquals("Kontoeinstellungen", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("English (US)")) {
        assertEquals("Account settings", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Español (ES)")) {
        assertEquals("Ajustes de cuenta", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Español (MX)")) {
        assertEquals("Ajustes de la cuenta", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Français")) {
        assertEquals("Paramètres du compte", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Italiano")) {
        assertEquals("Impostazioni account", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Português (BR)")) {
        assertEquals("Configurações da conta", accSettingsText);
    }else if(visibleTextLanguages.get(i).equals("Português (PT)")) {
        assertEquals("Definições da conta", accSettingsText);
}

```

```
}  
}  
}
```



Test Name: Test opting out of redesign test				
Description: Checks if you can still opt out of the redesign and return to the redesigned version				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Press the opt out button 2. Confirm the opt out 3. Assert if the design changed	Test Data:	Expected Result: The design changes to the older version	Actual Result: The actual result matches the expected result	PASS
Notes: We return to the new design, since all other tests are created with the new design in mind.				


```
@Test
void testOptOut() throws InterruptedException {
    WebDriver.get(baseUrl);

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

    WebElement buttonSwitch = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/
div[2]/div/div/div[2]/div[1]/div[11]/div[2]/div/button"))
    );
    buttonSwitch.click();

    // confirm opt out

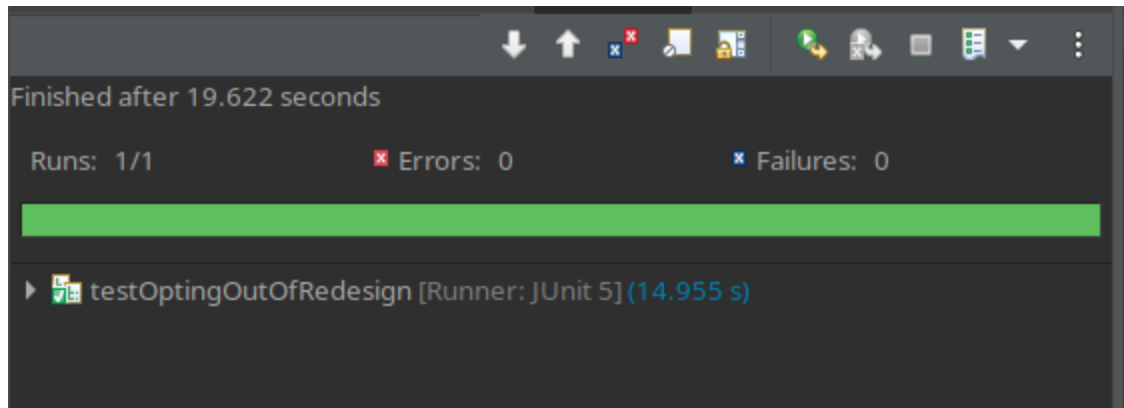
    wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/
div[4]/div/div/div/div/div/button[2]"))
    ).click();

    // check if design changed

    String msg = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[2]/div[1]/div/
button"))
    ).getText();

    assertEquals("GET NEW REDDIT", msg);

    Thread.sleep(3000);
}
```



Test Name: User report flow				
Description: Checks if the user is properly blocked				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached. A new non blocked username				
Test Steps: 1. Navigate to the profile page we want to block 2. Block the user 3. Refresh the page 4. Assert the user is blocked	Test Data: We hard code a user to block RightImage9461.	Expected Result: The user is successfully blocked.	Actual Result: The actual result is in line with the expected result.	PASS
Notes: For every instance the test is run, we need to supply it with a new non blocked username				

```

@Test
void testUserBlock() {
    webDriver.get(baseUrl);

    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

    WebElement buttonShowMore = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[2]/div/div[1]/div/div[7]/div/button"))
    );
    buttonShowMore.click();

    //block user
    WebElement blockButton = wait.until(

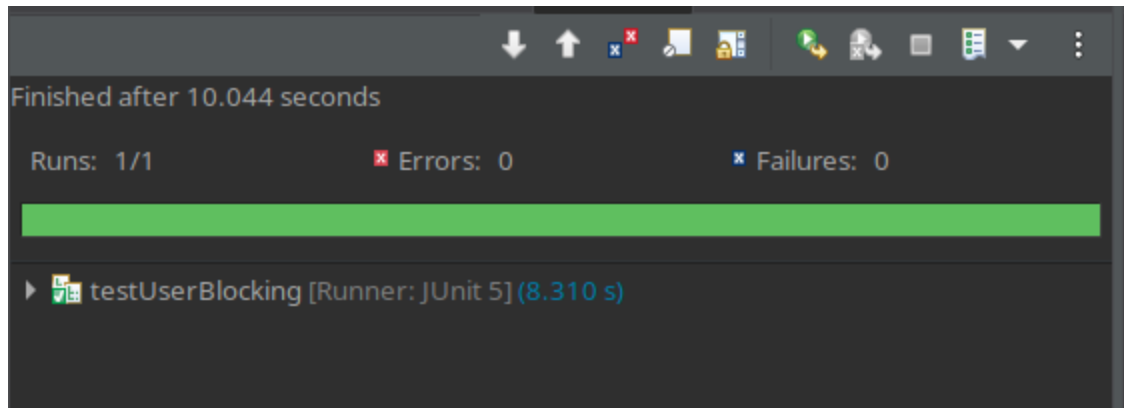
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[2]/div/div[1]/div/div[7]/button[1]"))
    );
    blockButton.click();

    webDriver.navigate().refresh();

    String s = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/h3"))
    ).getText();
    assertTrue(s.contains("is blocked"));
}

```



Test Name: Check Comment Persistence After ReSort

Description: Checks if the draft comment does not persist after filtering of comments

Pre-condition(s):

Test Steps:

1. Navigate to a post page
2. Scroll the comment form in view
3. Fill the comment
4. Resort comments
5. Accept alert
6. Assert if comment have been deleted

Test Data:

We use a particular post to perform the testing ("https://www.reddit.com/r/AskReddit/comments/zuckzx/what_does_europe_have_that_north_america_does_not/")

Expected

Result:
The comment draft is deleted after the page refreshes due to the modal alert

Actual Result:

The comment draft is not deleted after the page refresh

FAIL

Notes: This case deserves a further look into, while it is not a breaking change the flow is not consistent.

```

@Test
    void testCommentPersistance() throws InterruptedException {
        baseUrl =
"https://www.reddit.com/r/AskReddit/comments/zuckzx/what_does_europe_have_that_north_america_does_not/";
        webDriver.get(baseUrl);
        webDriver.manage().window().maximize();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

        WebElement commentForm = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div/div/div/div[2]/div/div[1]/div/div/div/div/div/div/div"))
        );
        JavascriptExecutor jse2 = (JavascriptExecutor) webDriver;
        jse2.executeScript("arguments[0].scrollIntoView", commentForm);
        commentForm.click();

        commentForm.sendKeys("Hello");

        webDriver.findElement(By.id("CommentSort--SortPicker")).click();

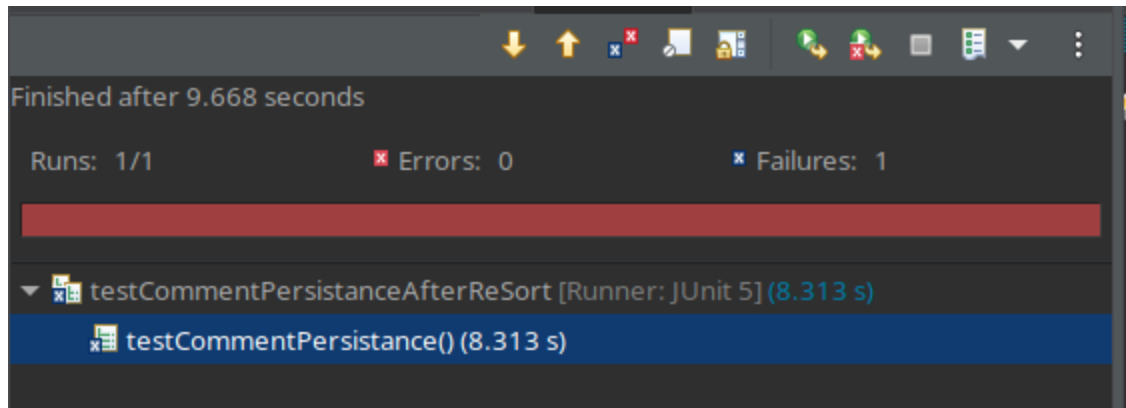
        wait.until(
            ExpectedConditions.visibilityOfElementLocated(By.linkText("Top"))
        ).click();

        webDriver.switchTo().alert().accept();

        String commentFormText =
webDriver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div/div/div/div[2]/div/div[1]/div/div/div/div/div/div/div/span/span")).getText();

        assertEquals(commentFormText, "Hello");
    }

```



3.3. Posts

Test Name: Test post draft				
Description: Test if post draft persits the draft				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Open post submit page 2. Fill in post 3. Save post as draft 4. Assert if draft is saved	Test Data: We hard code the post title as "Test Title" and post description as "Test Text"	Expected Result: The draft is successfully saved.	Actual Result: The actual result is the same as the expected result.	PASS
Notes:				

```

@Test
    void testPostDraftFunctionality() throws InterruptedException {
        webDriver.get(baseUrl);
        WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

        WebElement titleTextArea = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[2]/div[1]/div/textarea")))
        );

        titleTextArea.sendKeys("Test Title");
        WebElement textTextArea =
webDriver.findElement(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[2]/div[2]/div/div/div[3]/div/div[1]/div/div/div"));
        textTextArea.sendKeys("Test Text");

        WebElement saveDraftButton =
webDriver.findElement(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[3]/div[2]/div/div/div[2]/button"));

        saveDraftButton.click();
        Thread.sleep(400);

        // go to draft list
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[1]/div[2]/div[1]/button")))
        ).click();

        // check if draft is saved

        WebElement draftTitle = wait.until(

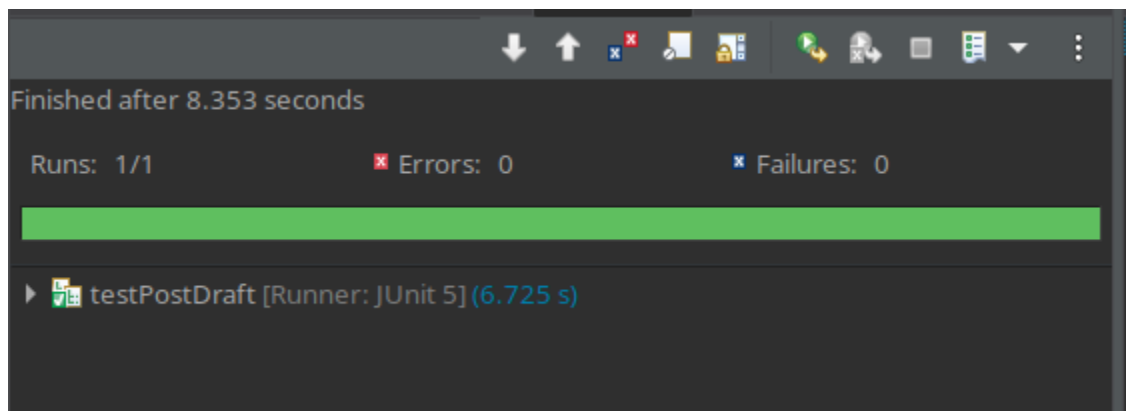
```

```

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[4]/div/div/div/div[2]/div[1]/div[2]/h2"))
    );
    System.out.println(draftTitle.getText());
    assertEquals(draftTitle.getText(),"EDITING: Test Title");

}

```



Test Name: Test post posting				
Description: Check if posts are created successfully.				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Open the submit post page 2. Fill in the post 3. Submit a post 4. Assert if post is created successfully.	Test Data: We hard code the post title as "Test Title" and post description as "Test Text"	Expected Result: The post is posted successfully.	Actual Result: The actual result is the same as the expected result.	PASS

Notes: The test only tests if the post was created by the cached user.

```
@Test
void testPostPosting() throws InterruptedException {
    webDriver.get(baseUrl);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

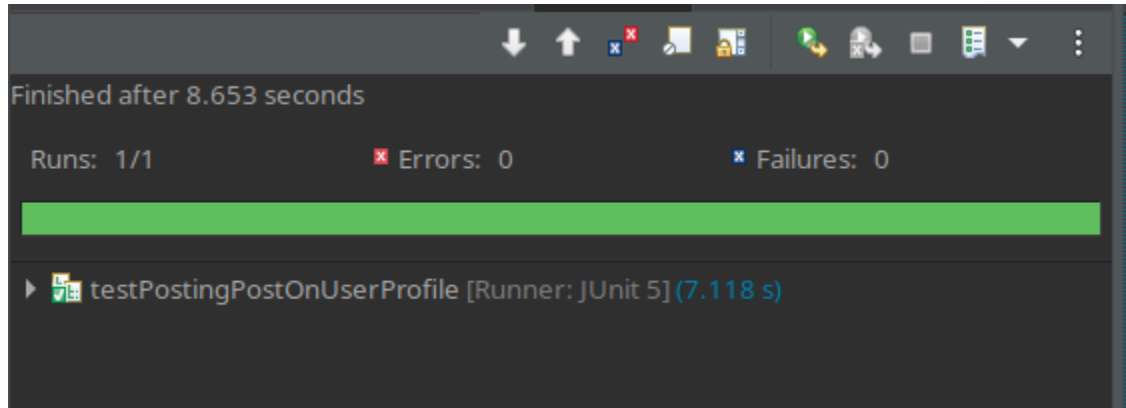
    WebElement titleTextArea = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"AppRouter-main-content\"]/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[2]/div[1]/div/textarea"))
    );
    titleTextArea.sendKeys("Test Title");
    //
    WebElement textTextArea = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[2]/div[2]/div/div/div[3]/div/div[1]/div/div/div"))
    );
    textTextArea.sendKeys("Test Text");

    webDriver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[3]/div[1]/div[2]/div[3]/div[3]/div[2]/div/div/div[1]/button")).click();
    //
    WebElement upt = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div/div[2]/div[2]/div[1]/div[3]/div[1]/div/div[2]/div/div[1]"))
    );

    System.out.println(upt.getText());
    //Thread.sleep(10000);
    assertEquals(upt.getText(), "Posted by\n")
}
```

```
+ "u/Character0k8020\n"
+ "just now");
```

```
}
```



3.4. Social links

Test Name: Test adding of social link				
Description: The test should add a social link redirect				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached				
Test Steps: 1. Navigate to profile page 2. Choose to add a link 3. Choose reddit as a link 4. Add a link 5. Assert that the link has been aded	Test Data: 3.0	Expected Result: The social redirect link is added.	Actual Result: The social redirect link is added.	PASS
Notes:				

```

class testAddSocialLinkRedirect {

    private static WebDriver webDriver;
    private static String baseURL;

    @BeforeAll
    static void setUp() {
        System.setProperty("webdriver.chrome.driver",
            "/home/musaka/Desktop/Semester 5/svvt/chromedriver");

        ChromeOptions options = new ChromeOptions();
        options.addArguments("--no-sandbox",
            "--disable-notifications",
            "--disable-extensions");
        String path = "/home/musaka/Desktop/RedditProfile";
        options.addArguments("--user-data-dir="+path);

        options.addArguments("--start-maximized");

        webDriver = new ChromeDriver(options);

        baseURL = "https://www.reddit.com/settings/profile";
    }

    @AfterAll
    static void tearDownAfterClass() throws Exception {
        webDriver.quit();
    }

    @Test
    void testSocialLink() throws InterruptedException {
        webDriver.get(baseURL);

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

        WebElement addSocialLinkDiv = wait.until(
            ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]
                /div[2]/div/div/div[2]/div[1]/div[3]/div[2]/div/div"))
        );

        addSocialLinkDiv.click();

        WebElement redditSocialLink = wait.until(
            ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]

```

```

]/div[4]/div/div/section/div/div[2]"))
        );

        redditSocialLink.click();

        WebElement socialLinkInput = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]
]/div[4]/div/div/section/div/input"))
        );

        socialLinkInput.sendKeys("r/Home");

        // /html/body/div[1]/div/div[2]/div[4]/div/div/section/div/div[2]
        Thread.sleep(3000);

        wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]
]/div[4]/div/div/section/header/div/div[3]/button"))
        ).click();

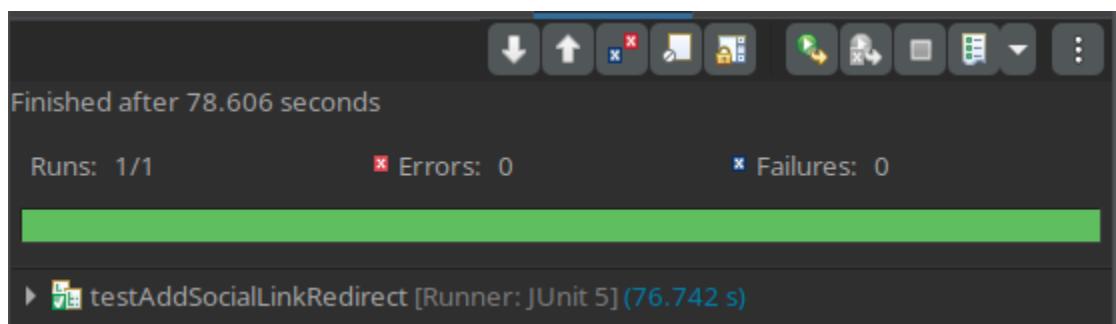
        webDriver.navigate().refresh();

        WebElement socialLink = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]
]/div[2]/div/div/div[2]/div[1]/div[3]/div[2]/div/div[1]"))
        );

        assertEquals("r/Home", socialLink.getText());

        Thread.sleep(3000);
    }
}

```



3.5. Miscellaneous

3.5.1. Testing of Emoji Send

Test Name: Testing of Emoji Send				
Description: The test should send a emoji to a user				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached. Also a username for the user to chat with.				
Test Steps: <ol style="list-style-type: none">1. Navigate to chat page2. Find the user to chat with3. Start Chat4. Send Emoji5. Assert Emoji is sent	Test Data: 3.0	Expected Result: The emoji is sent	Actual Result: The emoji is sent	PASS
Notes:				

```
@Test
void testUserBlock() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    WebElement l= webDriver.findElement(By.tagName("body"));
    String p = l.getAttribute("innerHTML");
    int numOfIcons =
this.countOfOccurrences(p,"https://www.redditstatic.com/desktop2x/img/snoom
oji/sloth.png");

    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));
```

```

        // i must find the user
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[1]/div[1]/div/button"))
        ).click();

        // send search attributes

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[1]/span/input"))
        ).sendKeys("muhasupa");
        // wait till search loads
        Thread.sleep(1000);
        // seelect user to send
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[2]/div"))
        ).click();
        // press start chat

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div[2]/div/button[2]"))
        ).click();

        // press emoji button
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div[1]/form/div/div/div/div[2]"))
        ).click();

        //press sloth emoji
        //
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/

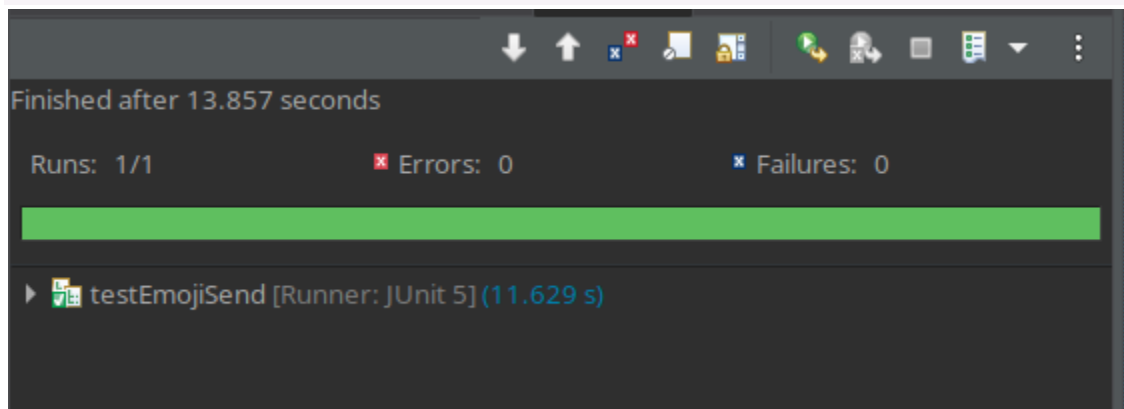
```

```

main/div[3]/div/div/div/div/div/div[34]"))
        ).click();
        Thread.sleep(2000);
        l= webDriver.findElement(By.tagName("body"));
        p = l.getAttribute("innerHTML");

        int numOfIconsAfter =
this.countOfOccurrences(p,"https://www.redditstatic.com/desktop2x/img/snoom
oji/sloth.png");
        assertEquals(numOfIconsAfter, numOfIcons+1);

```



3.5.2 Test Join Subreddit

Test Name: Testing Join Subreddit				
Description: The test should test if joining a subreddit works.				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached.				
Test Steps: 1. Navigate to subreddit 2. Join a subreddit	Test Data: 3.0	Expected Result: The user is added to the subreddit	Actual Result: The user is added to the subreddit	PASS
Notes:				

```
@Test
void testJoin() throws InterruptedException {
    webDriver.get(baseUrl);

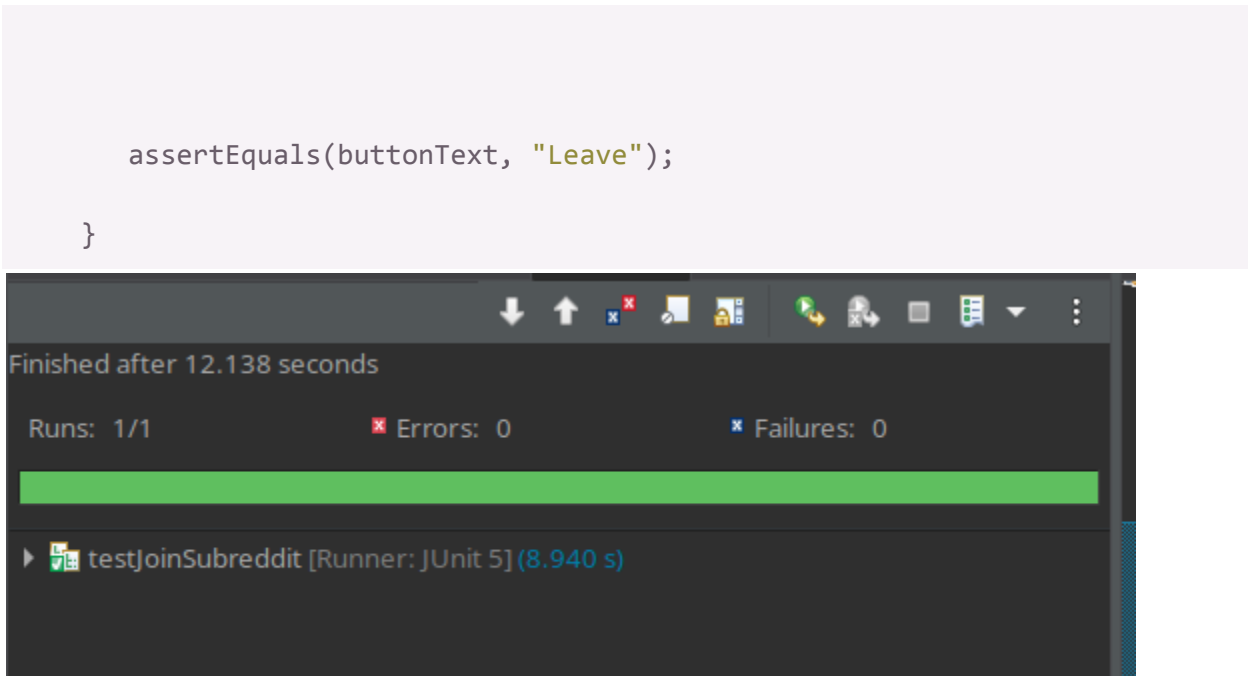
    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

    // i must find the user
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div/div[1]/div/div[2]/div/button
"))
        ).click();

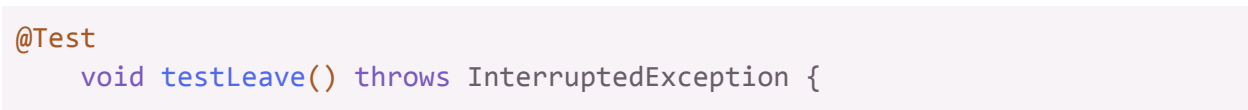
    String buttonText = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div/div[1]/div/div[2]/div[1]"))
        ).getText();
```

3.5.3. Test Leave Subreddit

Test Name: Testing Leaving Subreddit				
Description: The test should test if leaving a subreddit works.				
Pre-condition(s): 3.0 test case is a pre-condition, the chrome driver must have account credentials cached.				
Test Steps: 3. Navigate to subreddit 4. Leave a subreddit	Test Data: 3.0	Expected Result: The user leaves the subreddit.	Actual Result: The user leaves the subreddit	PASS
Notes:				



```

        WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

        // now i leave
        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div/div[1]/div/div[2]/div/button"))

                ).click();

        String buttonText = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/div/div[2]/div[1]/div/div[1]/div/div[2]/div[1]"))

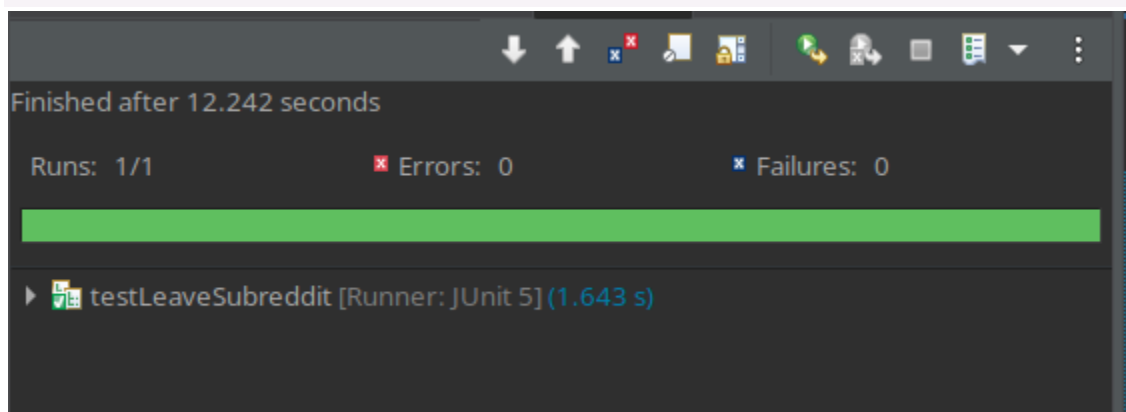
                ).getText();

        Thread.sleep(1500);

        assertEquals(buttonText, "Join");

    }

```



3.5.4 Test Login User Flow

Test Name: Testing Login User Flow				
Description: The test should test if user login flow works				
Pre-condition(s): The user must not be logged in, The user must have an account. The user must have access to their credentials.				
Test Steps: 1. Navigate to login page 2. Enter credentials 3. Log into Reddit	Test Data: Username and password credentials	Expected Result: The user logs into Reddit	Actual Result: The user logs into Reddit	PASS
Notes:				

```
@Test
void testUserBlock() throws InterruptedException {
    WebDriver.get(baseUrl);

    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

    WebElement u = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/main
/div[1]/div/div[2]/form/fieldset[1]/input"))
    );
    u.sendKeys("Single_Monitor_9475");

    WebElement p = wait.until(
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/main
/div[1]/div/div[2]/form/fieldset[2]/input"))
    );
    p.sendKeys("Sc8f&9fxwMfAcSPSuUt!V*5QFsgq");

    wait.until(
```

```

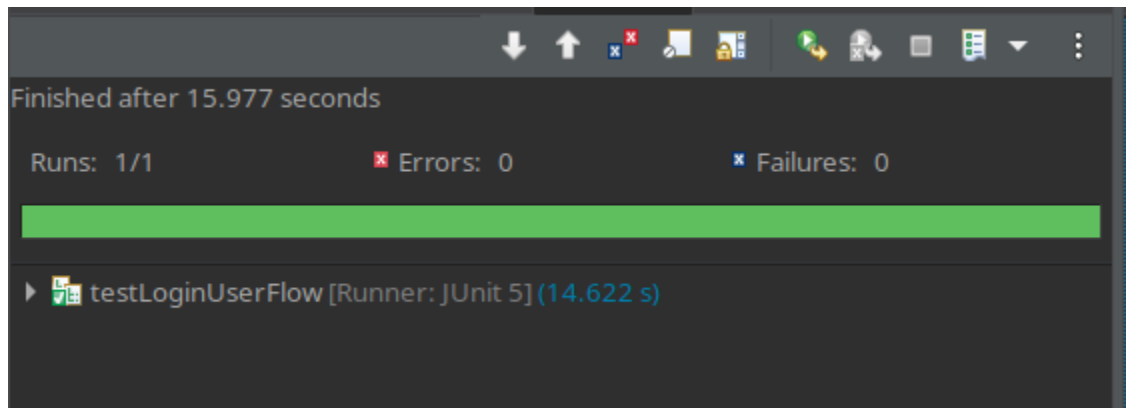
ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/main
/div[1]/div/div[2]/form/fieldset[5]/button"))
    ).click();

    String username = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div[1]/d
iv/div[2]/div[1]/header/div/div[2]/div[2]/div/div[2]/button/span[1]/span/sp
an"))
        ).getText();
    Thread.sleep(3000);
    assertEquals("Single_Monitor_9475\n" + "1 karma", username);

}

```



3.5.5 Test Message Delete

Test Name: Testing Message Deletion				
Description: The test should test if message is deleted				
Pre-condition(s): 3.0. There must be ONE message sent from the receiving user and no messages sent from the test driver.				
Test Steps: <ol style="list-style-type: none"> 1. Navigate to chat page 2. Find and send a message to user 3. Delete the message 	Test Data: An user to chat with	Expected Result: The sent message is deleted	Actual Result: The message is deleted	PASS
Notes:				

```

@Test
void testMessage() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);

    WebElement l= webDriver.findElement(By.tagName("body"));
    String p = l.getAttribute("innerHTML");
    int numOfIcons =
this.countOfOccurrences(p,"https://www.redditstatic.com/desktop2x/img/snoom
oji/sloth.png");

    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

    // i must find the user
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[1]/div[1]/div/button"))
        ).click();

    // send search attributes

    wait.until(

```

```

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[1]/span/input"))
        ).sendKeys("muhasupa");
    // wait till search loads
    Thread.sleep(1000);
    // seelect user to send
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[2]/div"))
        ).click();
    // press start chat

    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div[2]/div/button[2]"))
        ).click();

    WebElement input = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/form/div/div/textarea"))
        );
    input.clear();
    String salt = this.getSaltString();
    input.sendKeys(salt);
    // then i press the send button

    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/form/div/button"))
        ).click();

    // press the new message
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div[1]/div[1]/div/span[3]/span"))
        ).click();

```

```

        // press the delete button

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div[1]/div[1]/div/span[3]/div/div/div"))
                ).click();

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/div/div[2]/div/button[2]"))
                ).click();

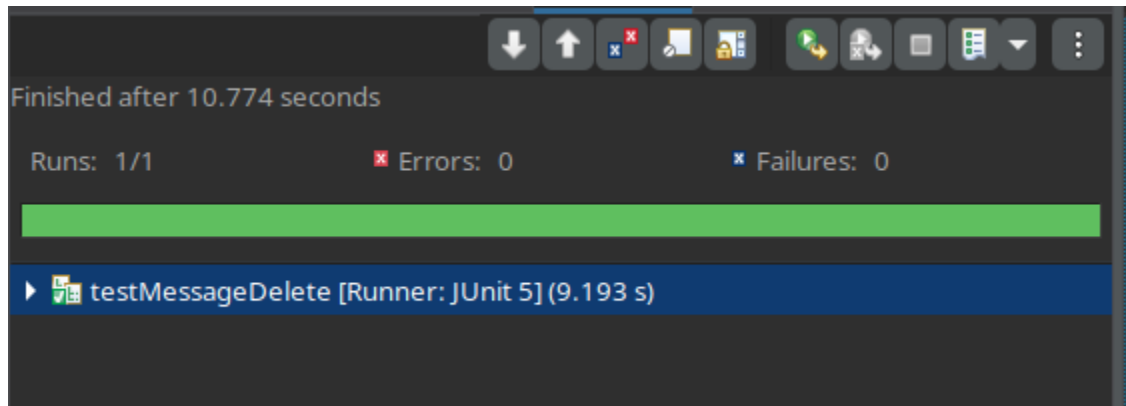
        String currentPageSource =
webdriver.findElement(By.tagName("body")).getAttribute("innerHTML");

        assertEquals(0, this.countOfOccurrences(salt, currentPageSource));

        Thread.sleep(300);

    }

```



3.5.6. Test Message Sending

Test Name: Testing Message Sending				
Description: The test should test if message is sent				
Pre-condition(s): 3.0.				
Test Steps: <ol style="list-style-type: none"> 1. Navigate to chat page 2. Send a message to user 	Test Data: An user to chat with	Expected Result: The message is sent	Actual Result: The message is sent	PASS
Notes:				

```

@Test
void testUserBlock() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebDriverWait wait = new WebDriverWait(webDriver,
Duration.ofSeconds(10));

    // i must find the user
    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[1]/div[1]/div/button"))
        ).click();

    // send search attributes

    wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[1]/span/input"))
        ).sendKeys("muhasupa");
    // wait till search loads
    Thread.sleep(1000);
    // seelect user to send
    wait.until(

```



```

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div/div[2]/div[2]/div"))
        ).click();
        // press start chat

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/div/form/div/div[2]/div/button[2]"))
        ).click();
        // now i send message to user
        WebElement input = wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/form/div/div/textarea"))
        );
        input.clear();
        String salt = this.getSaltString();
        input.sendKeys(salt);
        // then i press the send button

        wait.until(

ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/div/div/
main/div[1]/div[2]/div/form/div/button"))
        ).click();
        //check if page contains our salt
        String source = webDriver.getPageSource();
        assertTrue(source.contains(salt));

        Thread.sleep(5000);

}

```

