



Talan Academy

Rapport de fin de projet

Test Site web Decathlon.tn



Réalisé par :

Ayoub Ben Hadj Youssef

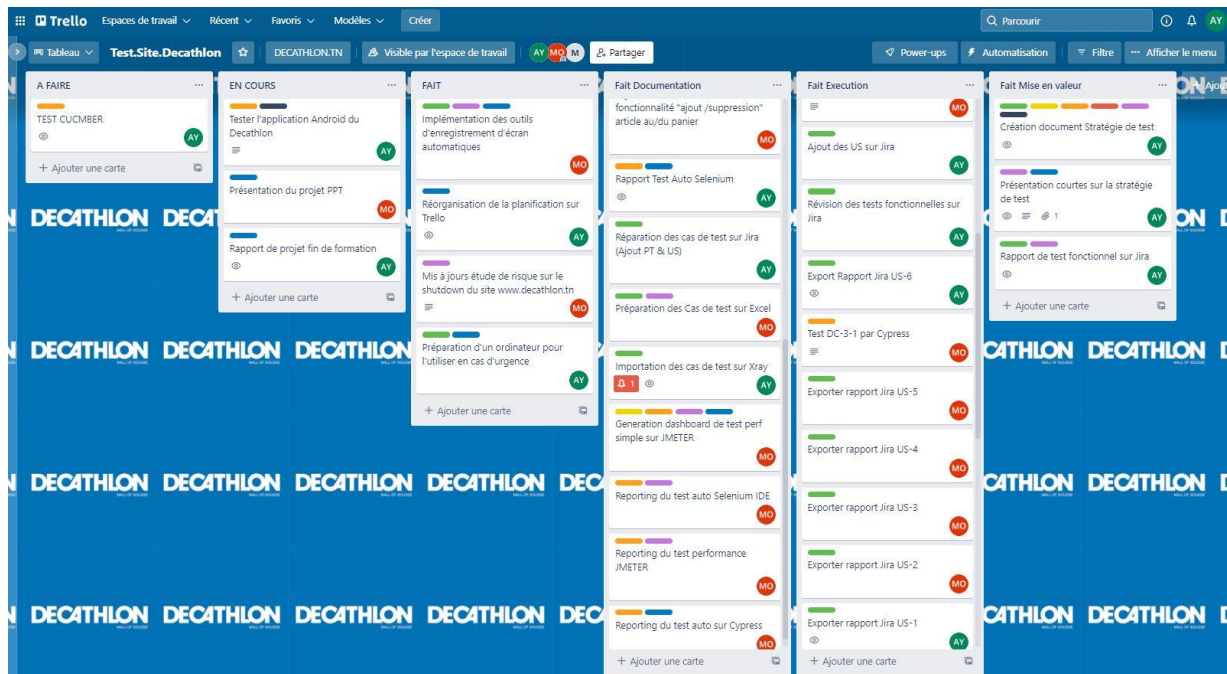
&

Mohamed Aziz El Ouni

METHODE DE TRAVAIL.....	3
TEST FONCTIONNEL	4
PERIMETRE D'INTERVENTION DE L'ACTIVITE	4
RECETTE ASSOCIEE AU PROJET	4
LES ELEMENTS DU PRODUIT/ LOGICIEL QUI SERONT TESTES.....	4
LES ELEMENTS QUI SONT EXCLUS DE LA STRATEGIE.....	4
IMPLEMENTATION DES TESTS	4
EXECUTION DES TESTS	5
CLOTURE DES TESTS	5
DIFFICULTES & CONTRAINTES TROUVES	5
TEST AUTO CYPRESS	6
ENVIRONNEMENT.....	6
SCENARIO DE TEST AUTOMATISE	6
CONTRAINTE	6
CONCEPTION DES CAS DE TEST	6
IMPLEMENTATION DES CAS DE TEST.....	6
EXECUTION DES CAS DE TEST.....	7
CONCLUSION.....	8
TEST AUTO SUR SELENIUM	8
PERIMETRE D'INTERVENTION DE L'ACTIVITE	8
ENVIRONNEMENT DE TEST.....	8
DIFFICULTES & CONTRAINTES TROUVES	9
TEST AUTO SELENIUM IDE	9
ENVIRONNEMENT.....	9
SCENARIO DE TEST AUTOMATISE	9
CONTRAINTE	9
CONCEPTION DES CAS DE TEST	9
EXECUTION DES CAS DE TEST.....	10
<i>Scénario Login avec informations valide.....</i>	<i>10</i>
<i>Scénario Login avec mail invalide</i>	<i>11</i>
<i>Scénario Login avec mot de passe invalide</i>	<i>12</i>
CONCLUSION.....	13
TEST DE PERFORMANCE.....	13
ENVIRONNEMENT.....	13
SCENARIO DE TEST ET DONNEES DE TEST.....	14
CONTRAINTE	14
IMPLEMENTATIONS DU TEST	14
EXECUTION DES CAS DE TEST.....	15
GENERATION DU RAPPORT DE TEST	15
CONCLUSION.....	17
CONCLUSION GENERALE	17

Méthode de travail

- Utiliser Trello pour la gestion des taches

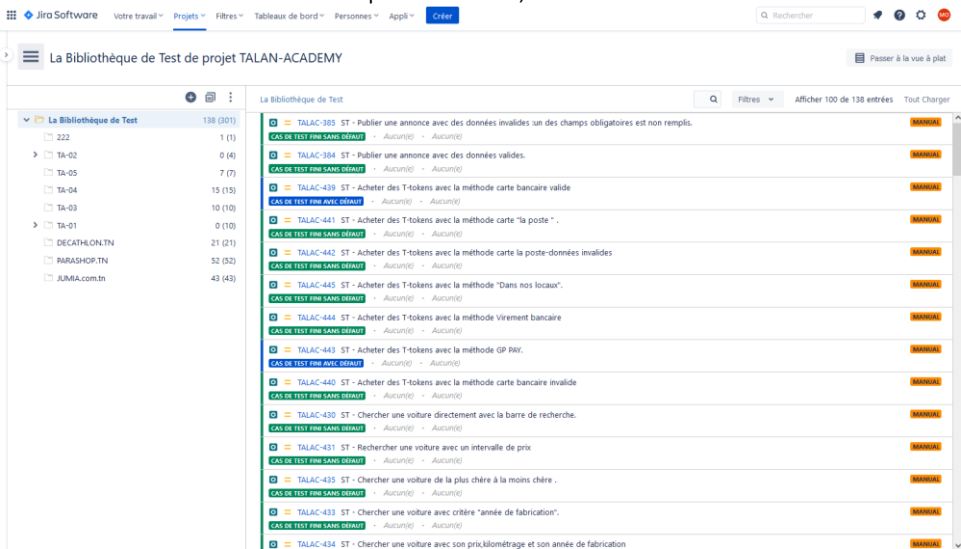


- Taches planifiées et sa répartition :

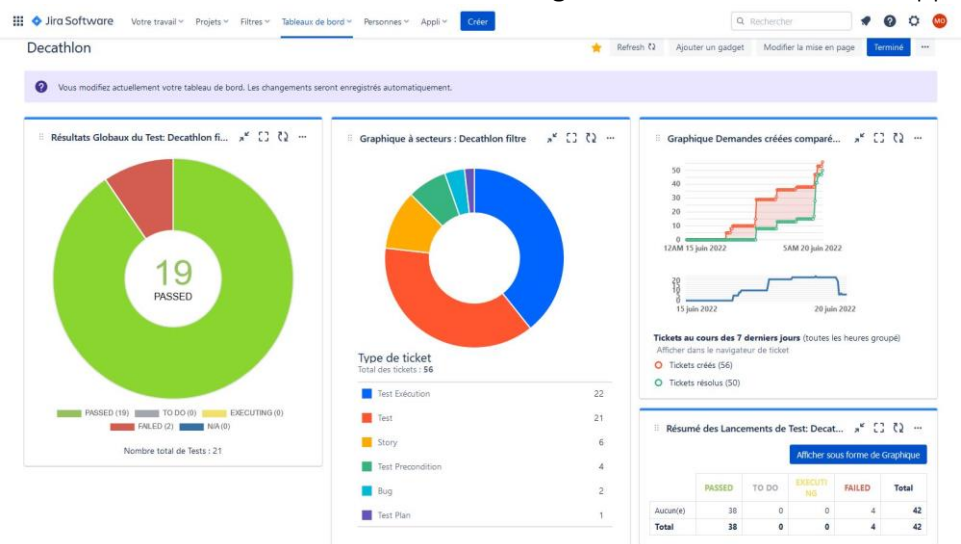
Taches	Répartition
Création de stratégie de test	Ayoub
Répartition des taches	Ayoub
Rédaction de recette de test	Mohamed Aziz
Révision et importation de la recette de test	Ayoub
Test US1 sur Jira	Mohamed Aziz
Test US2 sur Jira	Mohamed Aziz
Test US3 sur Jira	Ayoub
Test US4 sur Jira	Ayoub
Test US5 sur Jira	Ayoub
Test US6 sur Jira	Ayoub
Exporter le rapport de test fonctionnel	Ayoub
Création Dashboard des résultats sur Jira	Ayoub
Test auto de la fonctionnalité "Recherche par catégorie" sur Cypress	Mohamed Aziz
Test auto de la fonctionnalité "Accès au panier" sur Cypress	Mohamed Aziz
Test auto de la fonctionnalité "Recherche et sélection d'un article" sur Cypress	Mohamed Aziz
Test auto du fonctionnalité "Passage commande" sur Selenium	Ayoub
Test auto de la fonctionnalité "Remplir Message" sur Selenium	Ayoub
Test auto de la fonctionnalité "Modifier les infos personnelles" sur Selenium IDE	Mohamed Aziz

Exécution des tests

Une fois on a importé les cas de tests sur Jira, on vérifie les détails de chaque cas de test et on assure la connexion entre le plan de test, les US et les cas de tests.



On lance l'exécution des cas de tests et on génère le Dashboard et le rapport d'exécution



Les tests résultent deux anomalies d'un total de 21 cas de tests.

Clôture des tests

En vérifiant les critères de sortie, le clôture de test fonctionnel est validé. Les résultats de tests vont être vérifier par les outils test automatique.

Difficultés & Contraintes trouvés

Contrainte	Solution proposée
Importation des cas de test sur Xray doit être dans les normes demandés de l'outil	Suivre le rapport d'erreur et corriger
Vitesse d'exécution du test est importante Non-détection du l'élément suivant	Utiliser les outils "Wait" dans le code pour ralentir l'exécution.
Un seul projet pour tester de tous les sites	Créer un filtre par étiquette "Decathlon" afin de distinguer nos tickets

```

1  {
2    "errors": [
3      {
4        "elementNumber": 0,
5        "errors": [
6          {
7            "xray": "Test type not present"
8          }
9        ]
10     },
11     {
12       "elementNumber": 1,
13       "errors": [
14         {
15           "xray": "Test type not present"
16         }
17       ]
18     },
19     {
20       "elementNumber": 2,
21       "errors": [
22         {
23           "xray": "Test type not present"
24         }
25       ]
26     }
27   ]
28 }

```

Test Auto Cypress

Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

Scénario de test automatisé

- Accès au panier
- Recherche par catégorie
- Recherche et sélection d'un article

Contrainte

Contrainte	Solution proposée
Chute du site www.decathlon.tn	Lancer les scénarios sur www.decathlon.fr
Impossible d'afficher les résultats de générer un rapport sur le test effectué : Essai sur VSC effectué sans résultat favorable	Enregistrement par capture vidéo des scénarios de test et préparation d'un fichier avec les codes rédiger

Conception des cas de test

- Scénario accès au panier
- Scénario sélection par catégorie
- Scénario Recherche par catégorie

Implémentation des cas de test

A l'aide de l'outil Visual Studio Code, les cas de test ont été rédigé et implémenter

- Scénario Recherche par catégorie :

The screenshot shows the Visual Studio Code interface with the file explorer on the left and the editor on the right. The file explorer shows a project structure with a 'cypress' folder containing 'downloads', 'e2e', and 'fixtures' subfolders. The 'e2e' folder is expanded, showing several test files. The file 'recherche_par_categorie.cy.js' is selected and its content is displayed in the editor. The code is a Cypress test script for searching and selecting a category.

```
1 // Recherche et accès par catégorie
2 describe('select categorie "chaussures homme" on decathlon france', () => {
3   it('should select men categorie and choose "chaussures" ', () => {
4     cy.visit('www.decathlon.fr')
5     cy.get('#didomi-notice-agree-button > span').click();
6     cy.get('.menu-button').click();
7     cy.get('#cat-01 > span.svelte-78ehab').click();
8     cy.get('#cat-11 > span.svelte-1sqy87').click();
9     cy.get('[href="/browse/c0-homme/c1-chaussant/c2-chaussures/_/N-1f5113t"]').click();
10  })
11 })
```

- Scénario Recherche et sélection :

The screenshot shows the Visual Studio Code interface with the file explorer on the left and the editor on the right. The file explorer shows the 'e2e' folder expanded, and the file 'recherche_et_selection_dc.cy.js' is selected. The code is a Cypress test script for searching for an article and selecting a shoe.

```
1 // recherche et selection article
2 describe('search on decathlon france', () => {
3   it('should searcha chaussure on decathlon france site', () => {
4     cy.visit('www.decathlon.fr')
5     cy.get('#didomi-notice-agree-button > span').click();
6     cy.get('input.svelte-1ezbm7m').type('chaussure');
7     cy.get('button.svelte-1ezbm7m > .svg-icon').click();
8     cy.get('[data-supermodelid="168471"] > .dpb-models > div.svelte-1kvkbs3 > a.svelte-1kvkbs3 > .lz').click();
9   })
10 })
```

- Scénario Accès au panier :

The screenshot shows the Visual Studio Code interface with the file explorer on the left and the editor on the right. The file explorer shows the 'e2e' folder expanded, and the file 'accès_panier.cy.js' is selected. The code is a Cypress test script for accessing the shopping cart.

```
1 // accéder au panier
2 describe('access to shopping cart on decathlon france', () => {
3   it('should access to the shopping cart on decathlon france', () => {
4     cy.visit('www.decathlon.fr')
5     cy.get('#didomi-notice-agree-button > span').click();
6     cy.get('.cart-button').click();
7   })
8 })
```

Exécution des cas de test

L'exécution a été effectuée en lançant Cypress à partir du terminal de Visual Studio Code

(À regarder la vidéo dans le dossier partagé)

Conclusion

Malgré les contraintes rencontrées durant l'exécution de ces cas de test nous avons pu anticiper ces derniers grâce à une étude des risques et nous avons pu effectuer une couverture complète des cas de test implémenter.

Test auto sur Selenium

Périmètre d'intervention de l'activité

Ce test auto vise à valider les résultats de test fonctionnel des deux fonctionnalités suivantes :

1. Passage d'une commande : Recherche d'un article, Commander, Passer à une autre recherche d'article, choisir la quantité à acheter, et passer la commande.
2. Remplir un message : Passage à la page cible, Remplir les cases, Envoyer le message.

Environnement de test

Le code est rédigé sur IDE Eclipse en langage Java en ajoutant les bibliothèques Selenium et les Web drivers des navigateurs Chrome et Microsoft Edge.

- Code du Scénario "Remplir un message"

```
1 package Firstscript;
2
3 import java.time.Duration;
4
14 public class EnvoyerMessage {
15
16     public static void main(String[] args) throws InterruptedException {
17
18         // Provide path to Chromedriver
19         //System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");
20
21         // Provide path to Edgedriver
22         //System.setProperty("webdriver.chrome.driver", "C:\\msedgedriver.exe");
23         //System.setProperty("webdriver.ie.driver", "C:\\msedgedriver.exe");
24         //System.setProperty("webdriver.ie.driver", "C:\\msedgedriver.exe");
25         InternetExplorerOptions ieOptions = new InternetExplorerOptions();
26         ieOptions.attachToEdgeChrome();
27         ieOptions.withEdgeExecutablePath("C:\\Program Files (x86)\\Microsoft\\Edge\\Application\\msedge.exe");
28
29         // Instantiate Chromedriver class
30         WebDriver driver = new InternetExplorerDriver(ieOptions);
31
32         //ieOptions.attachToEdgeChrome();
33         driver.manage().window().maximize();
34
35         // Site access in Chrome browser
36         driver.get("https://www.decathlon.tn/");
37         driver.manage().timeouts().implicitlyWait(Duration.ofMillis(3000));
38
39         // Accés réseau magasins
40         driver.findElement(By.cssSelector("#header > div > div > div:nth-child(1) > div.col-md-6.text-right.nav--secondary > a > p")).click();
41
42         // Remplir un message
43         driver.findElement(By.cssSelector("#content > div > div.row.mb4 > div:nth-child(2) > section > form > section > div.form-group.row.mb4 > div > select")).sendKeys(Keys.chord("g"));
44         driver.findElement(By.name("name")).sendKeys("talan");
45         driver.findElement(By.name("lname")).sendKeys("Ayoub");
46         driver.findElement(By.name("email")).sendKeys("talanayoub@gmail.com");
47         driver.findElement(By.name("message")).sendKeys("Ce je suis en train de tester votre site web");
48         Thread.sleep(3000);
49         driver.findElement(By.name("submitMessage")).sendKeys(Keys.ENTER);
50     }
51
52 }
```

- Code du Scénario "Passage d'une commande"

```
1 import org.openqa.selenium.Keys;
2 import org.openqa.selenium.WebDriver;
3 import org.openqa.selenium.chrome.ChromeDriver;
4
5 public class PasserUneCommande {
6
7     public static void main(String[] args) throws InterruptedException {
8
9         // Provide path to Chromedriver
10        //System.setProperty("webdriver.chrome.driver", "C:\\chromedriver.exe");
11
12        // Instantiate Chromedriver class
13        WebDriver driver = new ChromeDriver();
14        driver.manage().window().maximize();
15
16        // Site access in Chrome browser
17        driver.get("https://www.decathlon.tn/");
18        driver.manage().timeouts().implicitlyWait(Duration.ofMillis(3000));
19
20        // Taper dans le recherche teste
21        driver.findElement(By.id("sa-search-input")).sendKeys("tente" + Keys.ENTER);
22
23        // Choix de la tente
24        driver.findElement(By.cssSelector("#hits > div > div > ol > li:nth-child(3) > div > a > img")).click();
25
26        // Ajout au panier
27        driver.findElement(By.cssSelector("#product_add_to_cart > div > div.pw.mb2.js-block-stock > div.add.js-add > button")).click();
28
29        // Switch to frame 3 success
30        driver.findElement(By.cssSelector("#blockcart-modal > div"));
31
32        // Cliquez en Continuer mes achats
33        driver.findElement(By.cssSelector("#blockcart-modal > div > div > div.modal-body > div.product-information > div.block_actions > button")).click();
34
35        // Recherche sac de couchage
36        driver.findElement(By.id("sa-search-input")).sendKeys("couchage");
37
38        // Thread.sleep(1000);
39        driver.findElement(By.id("sa-search-input")).sendKeys(Keys.ENTER);
40
41        // Choix du sac de couchage
42        //Thread.sleep(1000);
43        driver.findElement(By.cssSelector("#hits > div > div > ol > li:nth-child(3) > div > a > img")).click();
44
45        // Ajout du sac au panier
46        driver.findElement(By.cssSelector("#product_add_to_cart > div > div.pw.mb2.js-block-stock > div.add.js-add > button")).click();
47
48        // Switch to frame 3 success
49        driver.findElement(By.cssSelector("#blockcart-modal > div"));
50
51        // Accés au panier
52        driver.findElement(By.cssSelector("#blockcart-modal > div > div > div.modal-body > div.product-information > div.block_actions > div.checkout.cart-detailed-actions-btn > div > a")).click();
53
54        // Changer la quantité des sacs
55        driver.findElement(By.id("input-email")).sendKeys("talanayoub@gmail.com");
56
57        driver.findElement(By.id("input-email")).sendKeys(Keys.ENTER);
58
59        //Retour à la page précédente
60        driver.findElement(By.cssSelector("#app > nav > div > div:nth-child(1) > button > span")).click();
```


Difficultés & Contraintes trouvés

Contrainte	Solution proposée
Changement de la version du navigateur	Vérifier la version du navigateur avant d'exécuter
Vitesse d'exécution du test est importante Non-détection du l'élément suivant	Utiliser les outils "Wait" dans le code pour ralentir l'exécution.
Changement de Frame	Trouver la commande pour se basculer du Frame principal à un autre

Test Auto SELENIUM IDE

Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

Scénario de test automatisé

- Login avec informations Valide
- Login avec mail invalide
- Login avec mot de passe Invalide

Contrainte

Contrainte	Solution proposée
Impossibilité de générer des rapports de test après exécution → indisponibilité de l'extension test report (selenium IDE)	Enregistrement par capture vidéo de l'exécutions des scénarios de test et des Screenshot et préparation d'un fichier avec les codes rédiger

Conception des cas de test

Avec l'outil selenium IDE, nous avons enregistré les scénarios de test et réalisé un test de bout en bout.

Exécution des cas de test

Scénario Login avec informations valide

The screenshot displays the Selenium IDE interface for a project named "Campagne de test décatlon Tunisie". The test scenario "login_with_valid_informations*" is selected. The test steps are as follows:

Command	Target	Value
1 open	/	
2 set window size		1552x840
3 mouse over	css= account-button	
4 mouse out	css= account-button	
5 click	css= account-button_text	
6 click	css= account-dropdown_connect-button	
7 click	id=input-email	
8 type	id=input-email	elounimohamedaziz@gmail.com
9 click	id=lookup-btn	
10 type	id=input-password	Aziz1994
11 click	id=signin-button	

The log shows the following steps:

- 7. click on id=input-email OK
- 8. type on id=input-email with value elounimohamedaziz@gmail.com OK
- 9. click on id=lookup-btn OK
- 10. type on id=input-password with value Aziz1994 OK
- 11. click on id=signin-button OK

The final log entry states: "login_with_valid_informations" completed successfully.

Pour donner suite à l'exécution, un code est automatiquement généré :

```
public class LoginwithvalidinformationsTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void loginwithvalidinformations() {
        driver.get("https://www.decatlon.tn/");
        driver.manage().window().setSize(new Dimension(1552, 840));
        {
            WebElement element = driver.findElement(By.cssSelector(".account-button"));
            Actions builder = new Actions(driver);
            builder.moveToElement(element).perform();
        }
        {
            WebElement element = driver.findElement(By.tagName("body"));
            Actions builder = new Actions(driver);
            builder.moveToElement(element, 0, 0).perform();
        }
        driver.findElement(By.cssSelector(".account-button_text")).click();
        driver.findElement(By.cssSelector(".account-dropdown_connect-button")).click();
        driver.findElement(By.id("input-email")).click();
        driver.findElement(By.id("input-email")).sendKeys("elounimohamedaziz@gmail.com");
        driver.findElement(By.id("lookup-btn")).click();
        driver.findElement(By.id("input-password")).sendKeys("Aziz1994");
        driver.findElement(By.id("signin-button")).click();
    }
}
```

Scénario Login avec mail invalide

The screenshot shows the Selenium IDE interface for a project named "Campagne de test décaathlon Tunisie". The test scenario "login_with_invalid_mail" is selected. The test steps are as follows:

Step	Command	Target	Value
1	open	/	
2	set window size	1552x840	
3	click	css= account-button_text	
4	click	css= account-dropdown_connect-button	
5	click	id=input-email	
6	type	id=input-email	elounimohamedaziz@gmail.com
7	click	id=lookup-btn	

The log shows the execution of the test steps, with the final status being "login_with_invalid_mail" completed successfully.

À la suite de l'exécution, un code est automatiquement est généré :

```
Le mode restreint est destiné à la navigation de code sécurisé. Approuvez cette fenêtre pour activer toutes les fonctionnalités. Gérer En savoir plus

LoginwithvalidinformationsTest.java LoginwithinvalidmailTest.java X
C: > Users > maelouni > Desktop > Test Auto with Selenium IDE > login_invalid_mail > LoginwithinvalidmailTest.java > ...

17 import org.openqa.selenium.support.ui.ExpectedConditions;
18 import org.openqa.selenium.support.ui.WebDriverWait;
19 import org.openqa.selenium.JavascriptExecutor;
20 import org.openqa.selenium.Alert;
21 import org.openqa.selenium.Keys;
22 import java.util.*;
23 import java.net.MalformedURLException;
24 import java.net.URL;
25 public class LoginwithinvalidmailTest {
26     private WebDriver driver;
27     private Map<String, Object> vars;
28     JavascriptExecutor js;
29     @Before
30     public void setUp() {
31         driver = new ChromeDriver();
32         js = (JavascriptExecutor) driver;
33         vars = new HashMap<String, Object>();
34     }
35     @After
36     public void tearDown() {
37         driver.quit();
38     }
39     @Test
40     public void loginwithinvalidmail() {
41         driver.get("https://www.decathlon.tn/");
42         driver.manage().window().setSize(new Dimension(1552, 840));
43         driver.findElement(By.cssSelector(".account-button_text")).click();
44         driver.findElement(By.cssSelector(".account-dropdown_connect-button")).click();
45         driver.findElement(By.id("input-email")).click();
46         driver.findElement(By.id("input-email")).sendKeys("elounimohamedaziz@gmail.com");
47         driver.findElement(By.id("lookup-btn")).click();
48     }
49 }
50
```

Scénario Login avec mot de passe invalide

The screenshot shows the Selenium IDE interface with a project named "Campagne de test décaathlon Tunisie". The test scenario "login_invalid_password" is being executed. The interface displays a table of commands and their targets/values.

Command	Target	Value
1 open	/	
2 set window size	1552x840	
3 click	css=account-button_text	
4 click	css=account-dropdown_connect-button	
5 click	id=input-email	
6 type	id=input-email	elounimohamedaziz@gmail.com
7 click	id=lookup-btn	
8 type	id=input-password	123654789
9 click	id=signin-button	

Below the table, there are input fields for Command, Target, Value, and Description. At the bottom, a log shows the execution steps and their status (OK).

Log:

- 5. click on id=input-email OK
- 6. type on id=input-email with value elounimohamedaziz@gmail.com OK
- 7. click on id=lookup-btn OK
- 8. type on id=input-password with value 123654789 OK
- 9. click on id=signin-button OK

The test "login_invalid_password" completed successfully.

À la suite de l'exécution, un code est automatiquement généré :

The screenshot shows a Java code file named "LoginInvalidpasswordTest.java" in Visual Studio Code. The code is a Selenium test script for logging in with an invalid password.

```
C:\Users> maelouni > Desktop > Test Auto with Selenium IDE > login_with_invalid_password > LoginInvalidpasswordTest.java > ...
14 import org.openqa.selenium.Dimension;
15 import org.openqa.selenium.WebElement;
16 import org.openqa.selenium.interactions.Actions;
17 import org.openqa.selenium.support.ui.ExpectedConditions;
18 import org.openqa.selenium.support.ui.WebDriverWait;
19 import org.openqa.selenium.JavascriptExecutor;
20 import org.openqa.selenium.Alert;
21 import org.openqa.selenium.Keys;
22 import java.util.*;
23 import java.net.MalformedURLException;
24 import java.net.URL;
25 public class LoginInvalidpasswordTest {
26     private WebDriver driver;
27     private Map<String, Object> vars;
28     JavascriptExecutor js;
29     @Before
30     public void setUp() {
31         driver = new ChromeDriver();
32         js = (JavascriptExecutor) driver;
33         vars = new HashMap<String, Object>();
34     }
35     @After
36     public void tearDown() {
37         driver.quit();
38     }
39     @Test
40     public void loginInvalidpassword() {
41         driver.get("https://www.decaathlon.tn/");
42         driver.manage().window().setSize(new Dimension(1552, 840));
43         driver.findElement(By.cssSelector(".account-button_text")).click();
44         driver.findElement(By.cssSelector(".account-dropdown_connect-button")).click();
45         driver.findElement(By.id("input-email")).click();
46         driver.findElement(By.id("input-email")).sendKeys("elounimohamedaziz@gmail.com");
47         driver.findElement(By.id("lookup-btn")).click();
48         driver.findElement(By.id("input-password")).sendKeys("123654789");
49         driver.findElement(By.id("signin-button")).click();
50     }
51 }
```

Des vidéos de l'exécution ont été aussi enregistrés et seront remis avec les livrables

A la fin du test, Cypress génère un tableau récapitulatif comme suit

(Run Finished)

Spec		Tests	Passing	Failing	Pending	Skipped
✓ accès_categorie.cy.js	00:23	1	1	-	-	-
✗ accès_panier.cy.js	00:16	1	-	1	-	-
✓ recherche_et_selection_dc.cy.js	00:26	1	1	-	-	-
✓ todo.cy.js	00:05	6	6	-	-	-
✓ actions.cy.js	00:19	14	14	-	-	-
✓ aliasing.cy.js	00:04	2	2	-	-	-
✓ assertions.cy.js	00:05	9	9	-	-	-
✓ connectors.cy.js	00:03	8	8	-	-	-
✓ cookies.cy.js	00:03	5	5	-	-	-
✓ cypress_api.cy.js	00:05	12	12	-	-	-
✓ files.cy.js	00:02	4	4	-	-	-
✓ local_storage.cy.js	00:02	1	1	-	-	-
✓ location.cy.js	00:02	3	3	-	-	-
✓ misc.cy.js	00:04	6	6	-	-	-
✓ navigation.cy.js	00:04	3	3	-	-	-
✓ network_requests.cy.js	00:07	6	6	-	-	-
✓ querying.cy.js	00:03	5	5	-	-	-
✓ spies_stubs_clocks.cy.js	00:05	7	7	-	-	-
✓ traversal.cy.js	00:06	18	18	-	-	-
✓ utilities.cy.js	00:04	5	5	-	-	-
✓ viewport.cy.js	00:04	1	1	-	-	-
✓ waiting.cy.js	00:07	2	2	-	-	-
✓ window.cy.js	00:02	3	3	-	-	-
✗ 1 of 23 failed (4%)	02:51	123	122	1	-	-

Conclusion

Après enregistrement des scénarios, l'exécution a été effectuée avec succès malgré la contrainte rencontrés, la solution apportée a permis d'avoir une idée sur l'exécution de ces scénarios.

Pour les prochaines campagnes en utilisant Selenium IDE, un outil de reporting sera mis en place afin de faciliter la gestion des KPI.

Test de Performance

Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

Scénario de test et données de test

TEST DE PERFORMANCE DECATHLON TUNISIE	
	
Outil de test	<div>-Apache JMeter</div> <div>-Blazemeter</div> <div> </div>
Nombre d'utilisateur injecté	5 utilisateurs par secondes
Temp de sollicitations	Pendant 1 minutes
Scénario	Accéder au site www.decathlon.tn , réaliser une recherche sur un polo et ajouter l'article au panier

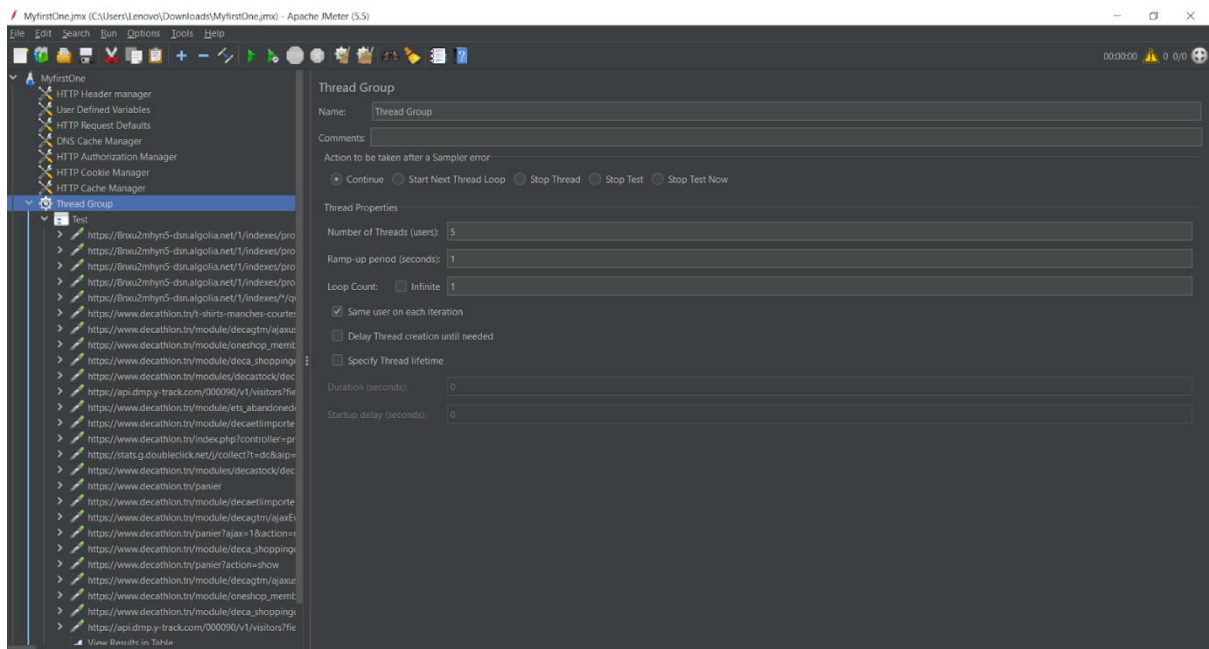
Contrainte

Contrainte	Solution proposée
Impossibilité de réaliser un test de performance en injectant un grand nombre d'utilisateur durant une période importante →Risque de blocage de la part du site → durée de génération de rapport trop longue	Réaliser un test de performance simple en diminuant le nombre d'utilisateur et le temps.

Implémentations du test

Rédaction du scénario à l'aide de l'extension chrome Blazemeter.

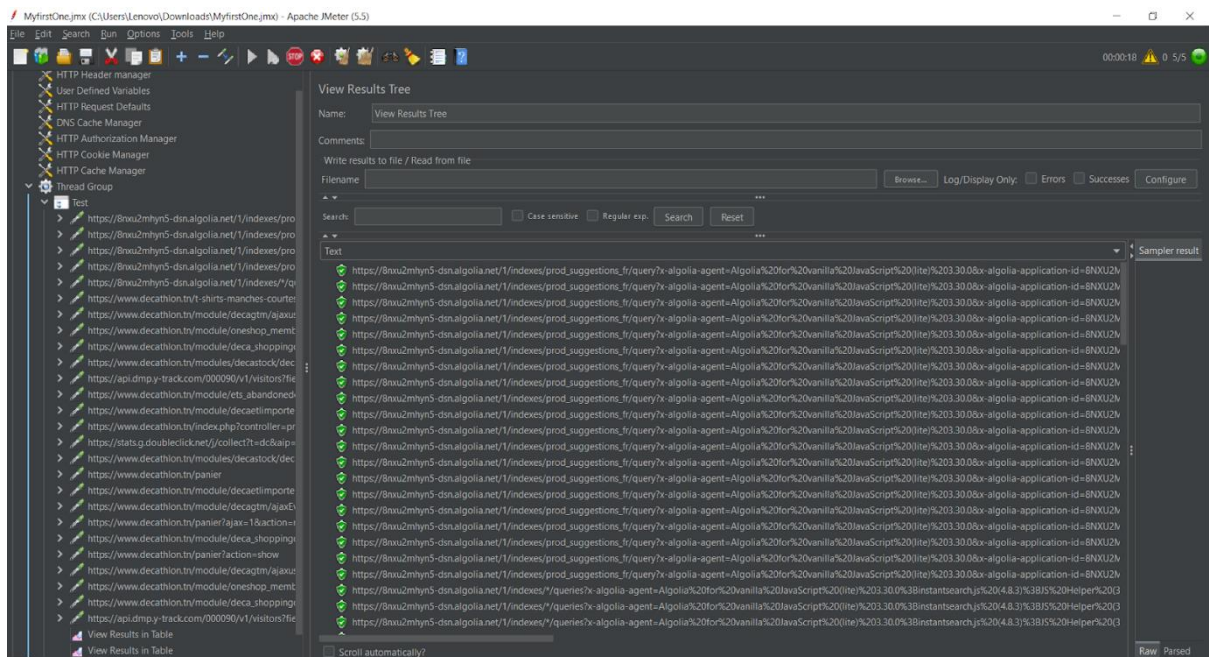
Implémenter le scénario enregistrer sur l'outil JMETER



Injection des paramètres de test pour les performances du site5 utilisateur chaque secondes pendant une minute.

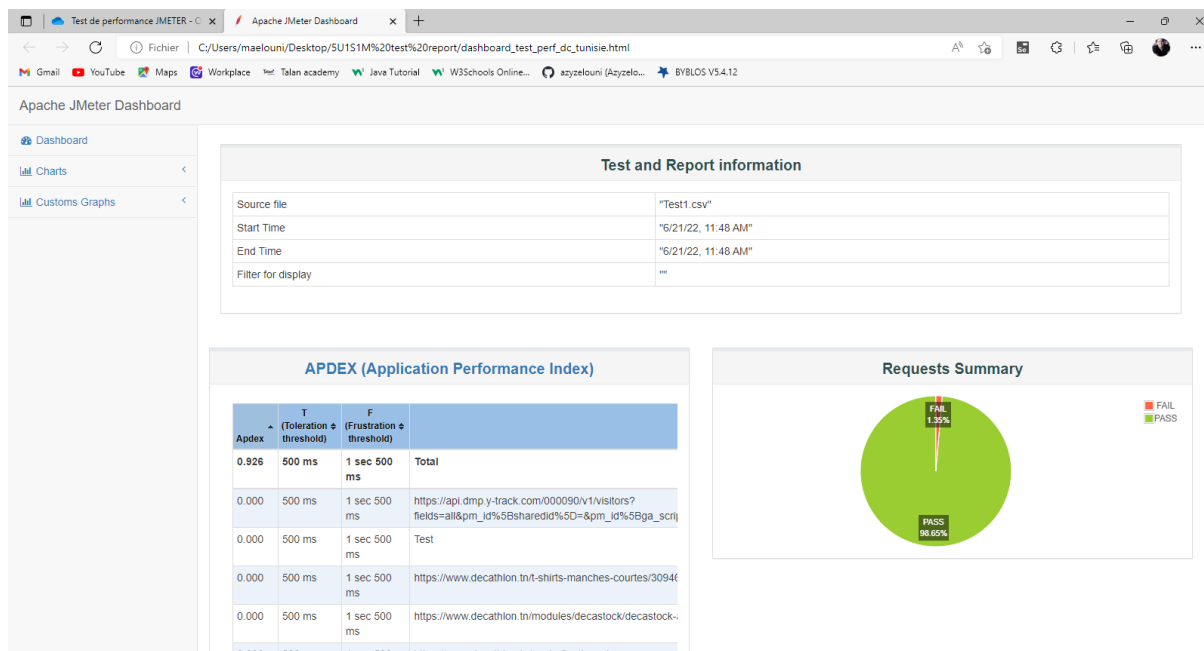
Exécution des cas de test

L'exécution a été réalisée à l'aide de l'outil JMeter après insertion des paramètres et choix de l'affichage grâce à l'ajout de la fonctionnalité "view results in tree" et "view results in table"



Génération du rapport de test

Le rapport a été généré grâce à l'outil invite de commande



Conclusion

Le test de performance sur le site [Decathlon.tn](https://www.decathlon.tn) a été réalisé de façon simple par suite de la période de génération des rapports qui posait contrainte pour l'avancement de la campagne et le temps estimé.

Un test plus robuste serait programmé lors d'un prochain sprint afin de bien tester la robustesse du site.

Conclusion générale

Ce projet a mis en test les fonctionnalités majeurs du site Decathlon.tn afin de vérifier son bon fonctionnement et améliorer l'expérience de l'utilisateur final du site.

Les tests sont effectués de la façon manuelle à travers l'outil de planification des tests manuels Xray Jira, et d'une façon automatique par les outils de tests automatique (Cypress, Selenium, et Selenium IDE). Un test de performance par l'outil Jmeter met l'accent sur la résistance du site à l'accès simultané par un grand nombre d'utilisateurs.

En outre, le test du délai de réponse et le test de l'application Android du site sont importants pour valider la fiabilité du service.