Plan de testare – Youtube

**Pregatirea Masinii Virtuale**

Soft-uri necesare:

-Oracle VM Virtual Box

-Microsoft Windows 10 iso

Pasii crearii masinii virtuale

1.

A screenshot of a computer

Description automatically generated

In casutele date se trag in felul urmator:

Name: orice nume

Folder: locatia unde se salveaza masina virtuala

Iso Image: locatia cu fisierul iso pentru windows 10

Type: Microsoft Windows

Version: Windows 10 (64-bit)

2. A screenshot of a computer

Description automatically generated

Slider-ul base memory e de preferat sa aibe minim 2GB

Numarul de procesoare alocate poate sa fie 1 pentru testare fara generare de media, 2 pentru testare cu generare foto si 3 pentru testare cu generare video

3. A screenshot of a computer

Description automatically generated

Spatiul recomandat pentru stoare este de 100GB pentru a instala toate soft-urile necesare testarii.

4. Se da next si finish si se porneste masina virtuala de pe butonul start

A screenshot of a computer

Description automatically generated

5. Instalarea de Windows este identica ca pe un calculator, se creeaza partitia C unde se aloca cei 100 GB si se instaleaza windows-ul

6. Dupa ce se instaleaza windows-ul, trebuiesc instalate Google Chrome, Visual Studio Community + Visual Studio Code, java jdk 1.8, eclipse si node.js.

Instalare si configurare Cypress:

1.Se creeaza un folder, se deschide command prompt si folosind comanda cd + locatia se intra in locatia folderului.

2.In command prompt se introduce commanda:

npm install cypress --save-dev

3.Pentru ca sa folosim xpath si sa generam loguri ale testarii, vom instala inca 2 plugin-uri cu urmatoarele comenzi:

npm install cypress-xpath

npm i --save-dev cypress-mochawesome-reporter

4.Se va deschide package.json folosind visual studio code si se vor adauga urmatoarele:

A screen shot of a computer program

Description automatically generated

5.In command prompt, folosim comanda:

npm run cypress:open

Asa deschidam prima data interfata cypress pentru debugging

6.Apasam pe end to end testing si configurare se face automat

7.Inchidem interfata si intram in cypress.config.js si adaugam urmatoarele:

A screen shot of a computer program

Description automatically generated

8. Deschidem cypress -> support -> e2e.js si adaugam urmatoarele:

A screen shot of a computer code

Description automatically generated

9. Acum se poate redeschide cypress cu comanda npm run cypress:open si sa se creeze testele si se pot edita in cypress -> e2e

Configurare Selenium

1. Se descarca fisierele selenium java, selenium server si chrome driver
2. Se creeaza un nou maven project in eclipse
3. Se alege group id org.apache.maven.archetypes artifact id maven-archetype-quickstart
4. Se alege un nume pentru group,artifact si package si se creeaza proiectul
5. Se da click dreapta pe proiect -> build path configuration
6. Pe tabul de libraries, apasam add external jars si adaugam selenium server jar si toate jar-urile din afara folderului lib
7. Apasam apply si dupa adaugam toate jar-urile din folderul lib
8. In pom.xml se adauga dependency-ul pentru reports si testng

<!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>5.1.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.9.0</version>

<scope>test</scope>

</dependency>

In caz ca doar dependency-ul nu functioneaza la TestNG, intrati pe eclipse marketplace si instalati-l ca plugin

1. Deoarece nu dorim sa avem ads pe videoclip-uri o sa cream un folder Extensions in care adaugam AdBlock.crx
2. La inceput de testare adaugam liniile

System.*setProperty*("webdriver.chrome.driver", "C:\\selenium webdriver\\ChromeDriver\\chromedriver.exe");

ChromeOptions opt = **new** ChromeOptions();

opt.addExtensions(**new** File("./Extensions/Adblock.crx"));

WebDriver driver = **new** ChromeDriver(opt);

Desigur locatia difera in functie de locul unde este chromedriver

Ce elemente se doresc a fi testate?

Pentru ca Youtube este o platforma sociala specializata pe videoclipuri o sa testam 2 pagini, home page si o pagina a unui videoclip.

In pagina de home se doreste a se verifica vizibilitatea elementelor si functionalitatea fiecarui element cu care se poate interactiona

In pagina videoclipului se doreste verificarea controalelor videoclipului, capacitatea de a se parcurge un videoclip cap coada. In pagina videoclipului, elementele secundare de UI nu sunt asa importante, functionalitatea de baza fiind videoclip-ul si controlul acestuia.

Pe langa aceste elemente se doresc si testarea login/logout, preluarea informatiilor in urma login-ului cum ar fi history, suggestion, youtube name, cat si deblocarea functionalitatilor posibile doar in urma login-ului cum ar fi subscribe, comment, like/dislike. Din nefericire aceste elemente nu pot fi verificate automat datorita politicilor youtube cu privire la autentificarea externa prin intermediul softului de debugging.

Test Case-uri:

**Testarea Volumului:**

1.Se acceseaza link-ul si se accepta cookie-urile

2.Se stocheaza valoarea initiala a volumului folosind API-ul youtube .getVolume()

3.Se schimba volumul

4.Se compara noul volum cu cel vechi

Rezultat asteptat:

Noul volum difera de cel vechi

**Testarea Butonului Play/Pause:**

1.Se acceseaza link-ul si se accepta cookie-urile

2.Se apasa butonul pause

3.Se preia folosind API-ul youtube .getPlayerState() daca videoclip-ul se afla in state-ul de paused

4.Se apasa butonul de play

5. Se preia folosind API-ul youtube .getPlayerState() daca videoclip-ul nu se mai afla in state-ul de paused

Rezultat asteptat:

Butonul de Play/Pause schimba state-ul video-ului in paused, respective playing

**Testarea butonului Next:**

1.Se acceseaza link-ul si se accepta cookie-urile

2.Se apasa butonul de next

3.Se compara noul url cu url-ul accesat

Rezultat asteptat:

Noul url difera de cel vechi, trecandu-se la alt videoclip

**Testarea vizibilitatii player-ului:**

1.Se acceseaza link-ul si se accepta cookie-urile

2.Se verifica ca video player-ul este vizibil

Rezultat asteptat:

Video player-ul este prezent si vizibil la accesarea link-ului

**Testarea AutoPlay On:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Autoplay-ul se verifica sa fie pornit (default on)
3. Se termina videoclip-ul si se asteapta 3 secunde pentru activarea functionalitatii
4. Se verifica ca noul url difera de cel vechi

Rezultat asteptat:

Videoclip-ul se termina cu success si se trece la urmatorul

**Testarea AutoPlay Off:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se opreste autoplay-ul (default on)
3. Se termina videoclip-ul si se asteapta 3 secunde pentru activarea functionalitatii
4. Se verifica ca url-ul nu s-a schimbat

Rezultat asteptat:

La finalul videoclip-ului, url-ul nu se schimba deoarece functia autoplay este oprita

**Testarea Play Speed:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se seteaza play speed-ul la orice valoare diferita de 1

Rezultat asteptat:

Videoclip-ul se misca mai repede/incet in functie de valoarea aleasa (mai incet daca e mai mic decat 1, respective mai repede daca este mai mare decat 1)

**Testarea Redarii Videoclipului:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se asteapta toata durata videoclipului

Rezultat asteptat:

Videoclip-ul se termina fara probleme/intreruperi

**Testarea Butonului de Subtitrari:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se apasa butonul de subtitrari
3. Se verifica ca subtitrarile au aparut
4. Se apasa iarasi butonul de subtitrari
5. Se verifica ca subtitrarile nu mai sunt vizibile

Rezultat asteptat:

La apasarea butonului, acesta isi schimba iconita sa indice activarea subtitrarilor si subtitrarile apar pe player-ul video, respectiv la re-apasarea butonului iconita acestuia indica oprirea subtitrarilor si acestea dispar de pe player-ul video

**Testarea Butonului de Miniplayer:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se apasa butonul de mini-player
3. Deoarece url-ul se schimba la apasarea butonului, se verifica url-ul si se compara cu cel vechi
4. Se apasa butonul de expand
5. Se verifica ca url-ul este acelasi cu url-ul accesat

Rezultat asteptat:

Butoanele de miniplayer si expand sunt functionale si miniplayer-ul este vizibil in dreapta jos

**Testarea Butonului History:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea si functionalitatea butonului history

Rezultat asteptat:

Butonul history este vizibil si te duce pe pagina [www.youtube.com/feed/history](http://www.youtube.com/feed/history)

**Testarea Butonului Subscriptions:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea si functionalitatea butonului Subscriptions

Rezultat asteptat:

Butonul history este vizibil si te duce pe pagina [www.youtube.com/subscriptions](http://www.youtube.com/subscriptions)

**Testarea Butonului Shorts:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea si functionalitatea butonului Shorts

Rezultat asteptat:

Butonul history este vizibil si te duce pe pagina [www.youtube.com/shorts](http://www.youtube.com/shorts)

**Testarea Butonului Home:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea si functionalitatea butonului Home

Rezultat asteptat:

Butonul history este vizibil si te duce pe pagina [www.youtube.com](http://www.youtube.com)

**Testarea Butonului You:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea si functionalitatea butonului You

Rezultat asteptat:

Butonul history este vizibil si te duce pe pagina [www.youtube.com/feed/you](http://www.youtube.com/feed/you)

**Testarea Logo-ului:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea logo-ului si se testeaza functionalitatea de home de pe acesta

Rezultat asteptat:

Logo-ul este vizibil si la apasarea pe acesta, user-ul este dus pe [www.youtube.com](http://www.youtube.com)

**Testarea Search Bar-ului:**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se verifica vizibilitatea search bar-ului
3. Se introduce un text si se apasa search
4. Se verifica vizibilitatea si relevanta rezultatelor

Rezultat asteptat:

Search Bar-ul este vizibil si rezultatele sunt relevante cu text-ul introdus si vizibile

**Testarea Sugestiilor de Search(necesita istoric):**

1. Se acceseaza link-ul si se accepta cookie-urile
2. Se introduce text partial
3. Se verifica vizibilitatea si relevanta sugestiilor

Rezultat asteptat:

Sugestiile sunt vizibile, iar apropierea si relevanta lor de text-ul introdus este descrescatoare de la prima sugestie la ultima, de sus in jos

**Testarea Sertarului Lateral:**

1.Se acceseaza link-ul si se accepta cookie-urile

2.Se apasa butonul pentru extend drawer

3.Se verifica vizibilitatea sertarului

4.Se apasa butonul de retreat drawer

5.Se verifica vizibilitatea side bar-ului din stanga

Rezultat asteptat:

Sertarul este vizibil si functional

**Citirea si interpretarea rezultatelor:**

**Cypress Results:**

A screenshot of a computer

Description automatically generated

A white line with black lines

Description automatically generated with medium confidence

**Selenium results:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Testele efectuate asupra platformei youtube sunt trecute, cum era de asteptat fiind un produs finisat si trecut prin testarea interna a companiei.

In rezultatele testarii UI-ului se pot observa timpi putin mai ridicati la task-uri simple, dar acestia se explica prin inducerea artificiala a unui timp de asteptare ce poate fi optimizat ulterior.

In rezultatele testarii Comenzilor Player-ului Video se pot observa timpi extrem de ridicati pentru comenzile AutoPlay On si AutoPlay Off, dar motivul este verificarea functiei prin introducerea de playback speed. Aceasta abordare a fost folosita pentru a conserva url-ul de test pentru toate testele. Pentru a optimiza aceasta parte, putem face un url parametrizat pe start la finalul videoclip-ului. Totusi un aspect pozitiv in acest test este ca se verifica si functionalitatea playback speed cat si capacitatea videoclipului de a rula start-end.

Totusi se poate observa ca, comparativ cu testele de UI, testele de comenzi sunt in medie mai lungi. Acest lucru se datoreaza urmatoarelor aspecte:

1. Videoclipurile de pe youtube, in inexistenta unui abonament, au anunturi aplicate pe player-ul video, lucru ce poate faila testarea. Pentru a combate acest lucru, browser-ul ruleaza folosind un adblocker care, in urma initializarii, deschide ferestre noi ce trebuiesc adresate.
2. Pe langa anunturi, pentru a evita erori datorate ecranului de accept/decline cookies, testul automat accepta cookie-urile la deschiderea link-ului
3. De asemenea, orice element din pagina trebuie afisat inainte de initializarea oricarei actiuni, iar pentru a evita erori de tipul “elementul nu a fost gasit” sau “nu se poate interactiona cu elementul”, se acorda un timp de asteptare pentru ca elementele sa fie reimprospatate dupa fiecare actiune, iar comenzile ce modifica multe elemente au o durata mai mare de asteptare cu scopul de a evita aceleasi erori, fapt ce mareste timpul de testare.

Timpul total de testare in Selenium este de 388s, adica 6 minute si 28 de secunde. Datorita acestui timp mare, testarea poate fi considerata ineficienta, dar adaugand in raport si elementele testate cat si posibile erori cauzate in incercarea de a reduce timpul de executie, consider ca eficienta este medie cu mentiunea ca se pot reduce aproximativ 2 minute prin optimizari, iar daca nu se doreste testarea integrala a derularii video, durata de test ar putea scadea cu aproximativ 3-4 minute.

Timpul total de testare in Cypress este de 3 minute si 20 de secunde. Discrepanta majora intre testarea aceasta si cea din selenium este ca cypress nu a avut nevoie sa gaseasca o solutie pentru anunturi, deci timpul de adresare a ferestrelor noi a fost redus la 0. De asemenea, la fel ca in selenium, timpul ar putea fi redus la aproximativ 1 minute daca nu s-ar parcurge tot videoclipul.