

Devoir 3:

Auteur: Basma kaanane

Numero : 300142701

Professeur : Dr. Andrew Forward

TA: Aymen Mhamdi

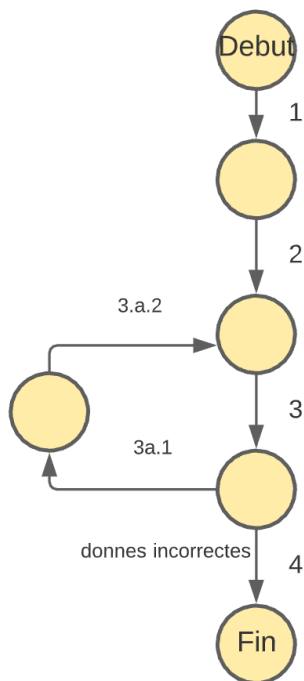
Question 1:

	Test positif	Test negative
F1	Un livre est ajoute au catalogue avec toute les information valide eg : categorie en science, id :abcdef, titre : X, nom de l'auteur Y, prix 10.5\$	Ajout d'un livre sans information ni categorie avec des entrées invalides eg : pas de categorie, ni de titre ni de nom d'auteur, id<5 ou >8, prix < 0.
F2.1	Rechercher un livre selon sa categorie eg : science	Recherche d'un livre d'une categorie inexistante et donc aucun livre n'est retourne
F.2.2	Recherche une valeur vide de categorie et donc tous les livre de cette categorie sont retournees	Recherche d'un livre d'une categorie inexistante et donc aucun livre n'est retourne
F3	Recherche d'un livre existant et ajout de celui-ci	Impossible d'ajouter un livre a cause d'une rupture de stock
F4	L'utilisateur est apte toutes les informations de sa commande lorsqu'il click sur le bouton shopping cart y compris le id, la description, le prix ainsi que la quantité	L'utilisateur n'est pas apte de voir le contenu de ses achats
F5	L'utilisateur peut changer la quantité de livres dans sa commande en ajoutant ou supprimant des livres	La quantite de livre dans la commande n'est pas mise a jour lorsque l'utilisateur modifie la commande (en supprimant ou ajoutant des livres)
F6	La date, le montant total, le montant des taxes et le montant des frais de manutention et d'expédition de la commande de livres d'un client sont calculés correctement	Un message d'erreur est affiche a la place d'afficher les donnees a cause de donnees invalides
F7	Un livre peut être supprime d'un catalogue lorsqu'il n'est pas vide	Impossibilité de supprimer un livre du catalogue a cause qu'il est vide
F8	L'admin peut se connecter correctement avec les donnees valides et peut ajouter ou supprimer des livres	L'utilisateur ne peut pas ajouter ou supprimer des livres a cause de donnees d'authentification incorrectes

Question 2 :

Cas de test 1 : Identification de l'administrateur

1) Graphe



Scénario principal

1. L'administrateur se rend sur la page de fonctionnement des administrateurs
2. e-BookStore lui demande ses identifiants de connexion.
3. L'administrateur saisit ses identifiants de connexion
4. e-BookStore présente la page d'opérations des administrateurs.

Alternatives

3. a. Identifiants de connexion erronés
3. a. 1. e-BookStore affiche un message d'erreur indiquant que les informations d'identification ne sont pas valides.
3. a. 2. 1) 3. a. 2. passer à l'étape 2

2) Scénarios assurant la couverture des branches du graphe

Id	Sequence	Description du scenario
1	1-2-3-4	l'administrateur se connecte avec des informations d'identification valides. (Le cours normal des événements)
2	1-2-3-3a.1-2-3-4	L'administrateur se connecte après avoir donné des informations d'identification invalides un nombre illimité de fois.

3) Cas de tests abstraits

Test Case: TC1_SignIn

Goal: Testez le déroulement principal de authentication

Scenario #: 1

Setup: Ouvrir la session page de connexion admin avec le nom "admin" et mot de passe « password »

Course of test case

	External Event	Reaction
1	L'utilisateur entre username=admin, password=password	La librairie est connecte en tant qu' admin et est capable d'administrer les operations

Success Criteria: Admin logged in

Test Case: TC2_SignIn

Goal: Test for successful admin login after a few failures entering credential.

Scenario #: 2

Setup: Opened the admin login page; account with name "admin" and password "password" exists.

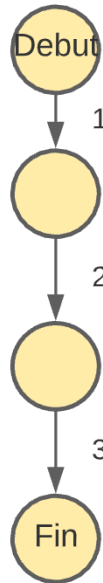
Course of test case

	External Event	Reaction
1	L'utilisateur entre username=admin, password=abcdef	Des messages d'erreurs sont affiches a cause des entres invalides

Success Criteria: Admin logged in

Cas de test 2 : deconnexion l'administrateur

1)Graphe



Scénario principal

1. L'administrateur choisit de se déconnecter
2. la librairie reconnaît que l'administrateur s'est déconnecté.

Ajouté

3. L'utilisateur retourne à l'écran de connexion

2) Scénarios assurant la couverture des branches du graphe

Id	Sequence	Description du scenario
1	1-2-3	Déroulement normal des événements. L'administrateur se déconnecte.

3) Cas de tests abstraits

Test Case: TC1_deconnexion

Goal: Testez le déroulement principal de la déconnexion.

Scenario #: 1

Setup: L'administrateur est connecté sur la page d'accueil

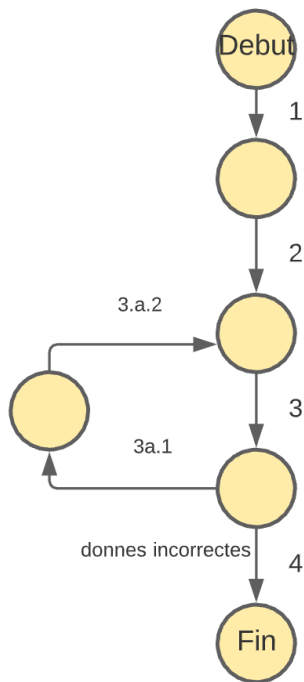
Course of test case

	External Event	Reaction
1	L'utilisateur clique sur le bouton "Déconnexion"	Le système déconnecte l'utilisateur et le redirige vers l'écran de connexion.

Success Criteria: Admin logged out

Cas de test 3 : Ajout d'un livre

1) Graphe



Scénario principal

1. L'administrateur entre les informations sur le livre
2. la librairie reconnaît l'ajout du livre au catalogue.

Alternatives

1. a. Informations erronées ou manquantes
 1. a. 1. la librairie affiche les messages d'erreur pertinents. 1. a. 2.
 1. a. 2. passer à l'étape 1
1. b. Identité du livre en double
 1. b. 1. la librairie affiche un message d'erreur d'identification de livre en double.
 1. b. 2. Passez à l'étape 1

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	L'administrateur est connecté et a ajouté un livre avec le droit et aucune information manquante
2	1-1a.1-1.a-2	Le livre ajouté est avec erronée ou manquante informations erronées ou manquantes
3	1-1b.1-1*-2	L'identifiant du livre saisi est dupliqué.
4	1-1a.1-1-1b.1-1-2	Livre ajouté après le rejet d'informations invalides avec un id invalide rejeté une fois.

3) Cas de tests abstraits

Test Case: TC1_ajoutLivre

Goal: Test du cours normal des événements pour ajouter des livres.

Scenario #: 1

Setup: L'administrateur est connecté ainsi que des informations valides ont été saisies.

Course of test case

	External Event	Reaction
1	L'administrateur saisit toutes les informations informations et clique sur le bouton ajouter	System adds the book to the catalogue and Bookstore shows successful added message

Success Criteria: Book added to catalogue.

Test Case: TC2_ajoutLivre

Goal: Test adding a book after entering invalid information.

Scenario #: 2

Setup: Admin is logged in; no information entered.

Course of test case

	External Event	Reaction
1	L'administrateur saisit des informations erronées informations et clique sur le bouton d'ajout	La librairie affiche un message d'échec

Success Criteria: Book added to catalogue.

Test Case: TC3_ajoutLivre

Goal: Test de l'ajout d'un livre après avoir entré des ID en double

Scenario #: 3

Setup: la librairie est en marche l'utilisateur est connecté en tant qu'administrateur.

Course of test case

	External Event	Reaction
1	L'administrateur clique sur le bouton Ajouter	1 Admin clique sur le bouton Ajouter Le système donne l'erreur "id already exists".

Success Criteria: Book added to catalogue.

Test Case: TC4_ajoutLivre

Goal: Test d'ajout d'un livre après la saisie d'informations non valides et d'un ID dupliqué.

Scenario #: 4

Setup: L'administrateur est connecté et aucune information n'a été saisi

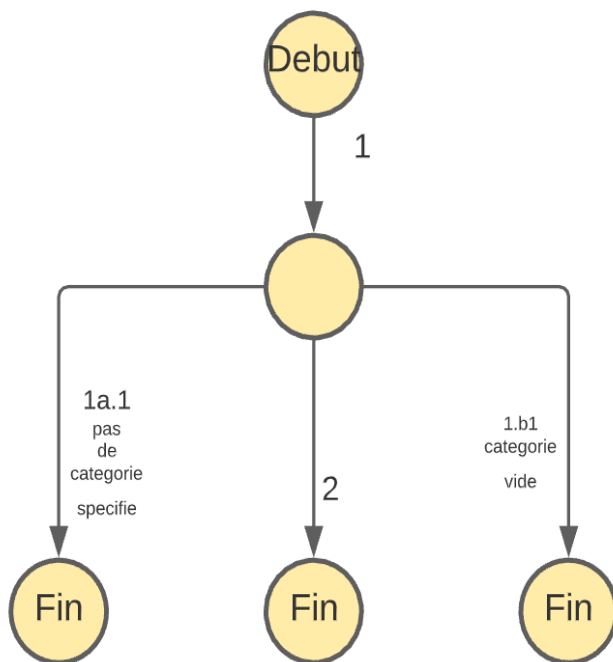
Course of test case

	External Event	Reaction
1	L'administrateur clique sur le bouton Ajouter	Le systeme affiche des messages d'erreur
3	Admin entre "aaaaa" en id et click sur le bouton ajout	Le systeme ajoute le livre dans le catalogue

Success Criteria: Book added to catalogue.

Cas de test 4 : Parcourir le catalogue

1) Graphe



Scénario principal

1. L'utilisateur spécifie une catégorie sur laquelle une recherche dans le catalogue doit être effectuée.
2. la librairie affiche une liste de livres correspondant à la catégorie sélectionnée.

Alternatives

1. a. Aucune catégorie n'a été spécifiée
 1. a. 1. la librairie affiche une liste de tous les livres du catalogue.
 1. b. Aucun livre correspondant à la catégorie recherchée n'a été trouvé.
 1. b. 1. la librairie affiche un message indiquant qu'aucun livre correspondant n'a été trouvé.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	L'utilisateur saisit une catégorie et clique sur le bouton de recherche puis une liste de livres s'affiche
2	1-1a.1	L'utilisateur n'entre pas dans une spécifier la catégorie et cliquer sur bouton de recherche, la liste de tous les livres s'affiche
3	1-1b.1	L'utilisateur saisit une catégorie et clique sur le bouton de recherche mais aucun livre n'est trouvé

3) Cas de tests abstraits

Test Case: TC1_recherche

Goal: Recherche normal d'un livre anu ne categorie

Scenario #: 1

Setup: AddBookTC1

Course of test case

	External Event	Reaction
1	L'utilisateur entre une certaine catégorie et clique sur le bouton de recherche bouton	La librairie montre tous les livres appartenant à la catégorie que l'utilisateur a saisi

Success Criteria: Books displayed

Test Case: TC2_rechercher

Goal: Tester le déroulement normal de l'affichage du catalogue complet

Scenario #: 2

Setup: None

Course of test case

	External Event	Reaction
1	L'utilisateur Entre une certaine catégorie et clique sur le bouton de recherche	La librairie affiche tous les livres

Success Criteria: All books displayed

Test Case: BrowseTC3

Goal: Test pour la recherche d'une catégorie vide/non existante **Scenario #:** 3

Setup: None

Course of test case

	External Event	Reaction
1	L'utilisateur saisit une catégorie spécifique avec aucun livre satisfait et clique sur le bouton de recherche bouton	La librairie affiche le message "aucun livre trouvé". message

Success Criteria: No books displayed

Cas de test 5 : Retirer le livre

1) Graphe



Scénario principal

1. L'administrateur choisit de supprimer un livre
2. la librairie affiche la liste des livres sans le livre supprimé.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Normal course of events. Book removed.

3) Cas de tests abstraits

Test Case: TC1_retirerLivre

Goal: Test the normal course of events for removing books

Scenario #: 1

Setup: Logged in as admin; non-empty category searched for

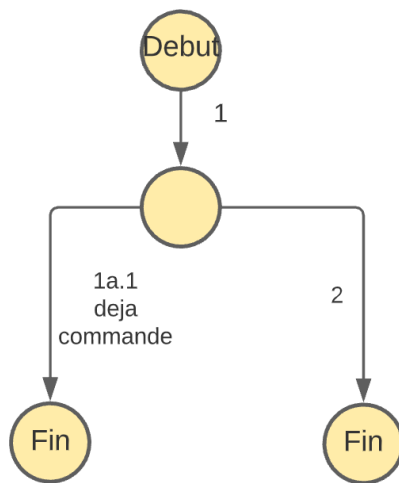
Course of test case

	External Event	Reaction
1	L'administrateur clique sur le bouton de suppression d'un livre.	La librairie affiche la liste des livres sans le livre supprimé

Success Criteria: Given book removed from catalogue

Cas de test 6 : Commander le livre

1) Graphe



Scénario principal

1. Le client sélectionne un livre à ajouter à sa commande de livres.
2. la librairie ajoute le livre à la commande de livres.

Alternatives

1. a. Le livre sélectionné figure déjà dans la commande de livres du client.
 1. a. 1. la librairie incrémente le nombre d'exemplaires du livre dans la commande de livres.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Le client sélectionne un livre et l'ajoute à sa commande.
2	1-1a.1	Le client a ajouté une commande dupliquée alors la librairie augmente le nombre de quantité de 1

3) Cas de tests abstraits

Test Case: TC1_commander

Goal: Tester le déroulement normal de l'ajout de livres à une commande

Scenario #: 1

Setup: Non connecté en tant qu'administrateur, la liste des livres est affichée

Course of test case

	External Event	Reaction
1	L'utilisateur sélectionne des livres en cliquant sur le bouton	La librairie ajoute le livre à la commande

Success Criteria: Book added to order

Test Case: TC2_commander

Goal: Test d'ajout de plusieurs livres à une commande à partir de la liste des résultats de la recherche

Scenario #: 2

Setup: Non connecté en tant qu'administrateur, la liste des livres est affichée

Course of test case

	External Event	Reaction
1	Utilisateur click sur "add to cart" for book ID'd " hall001" twice	System adds book with ID "hall001" to order a second time.

Success Criteria: Book added to order twice

Cas de test 7 : Voir la commande de livres

1) Grappe



Scénario principal

1. Le client choisit de visualiser sa commande de livres
2. la librairie affiche les informations relatives à la commande de livres.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Le client clique sur le bouton du panier après avoir ajouté livres dans sa commande.

3) Cas de tests abstraits

Test Case: TC1_voirCommande

Goal: Test the normal course of events for viewing a book order

Scenario #: 1

Setup: OrderBookTC1

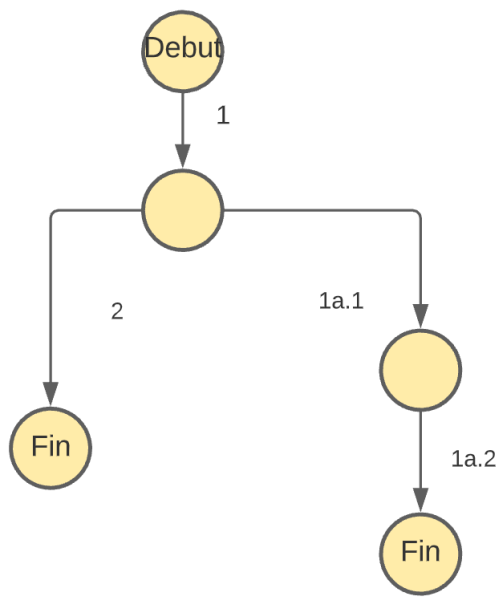
Course of test case

	External Event	Reaction
1	L'utilisateur clique sur le bouton panier	La librairie affiche la liste des livres dans sa commande

Success Criteria: bookstore is showing a list of books in customer's order.

Cas de test 8 : Mise à jour de la commande de livres

1) Grappe



Scénario principal

1. Le client spécifie un nombre d'exemplaires pour un livre commandé.
2. la librairie met à jour le coût total du livre.

Alternatives

1. a. Le nouveau nombre d'exemplaires est inférieur ou égal à 0.
 1. a. 1. la librairie met à jour le coût total du livre à 0
 1. a. 2. la librairie supprime le livre de la commande de livres.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Le client édite le nombre d'exemplaires du livre et la librairie met à jour le coût total
2	1-1a.1-1a.2	Le nombre d'exemplaires que le client saisi par le client est inférieur ou égal à 0. La librairie met à jour le coût total et supprime le livre

3) Cas de tests abstraits

4) **Test Case:** TC1_Miseajour

Goal: Testez le déroulement normal de la mise à jour du nombre d'exemplaires d'une commande.

Scenario #: 1

Setup: TC1_commander; TC1_voirCommande

Course of test case

	External Event	Reaction
1	L'utilisateur édite le nombre de livres	La librairie met à jour le coût total

Success Criteria: Order is updated

Test Case: TC2_Miseajour

Goal: Test de suppression d'un livre d'une commande en mettant à jour son numéro d'exemplaire à 0

Scenario #: 2

Setup: OrderBookTC1; ViewOrderTC1

Course of test case

	External Event	Reaction
1	L'utilisateur modifie le nombre de livres en moins à 0	La librairie met à jour le coût total à 0 et retire le livre de la commande

Success Criteria: Order list is emptied

Cas de test 9 : Verification de la commande en passant a la caisse

1) Graphe



Scénario principal

1. Le client choisit de passer à la caisse
2. la librairie traite la commande de livres et affiche les informations.

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Le client clique sur le bouton de paiement après avoir ajouté livres dans sa commande

3) Cas de tests abstraits

Test Case: CheckOutTC1

Goal: Test the normal course of action for checking out.

Scenario #: 1

Setup: OrderBookTC1; ViewOrderTC1; UpdateOrderTC1

Course of test case

	External Event	Reaction
1	L'utilisateur clique sur le bouton de paiement	La librairie traite la commande et affiche les informations relatives au paiement

Success Criteria: bookstore is showing the payment information of the customer's order.

Cas de test 10 : Sélection de langage

1) Graphe



Scénario principal

1. Le client sélectionne une langue

2. la librairie affiche des informations dans la langue sélectionnée

2) Scénarios assurant la couverture des branches du graphe

ID	Event Sequence	Scenario Description
1	1-2	Le client clique sur le bouton de paiement après avoir ajouté livres dans sa commande

3) Cas de tests abstraits

Test Case: SelectLanguageTC1

Goal: Test the normal course of events for changing languages.

Scenario #: 1

Setup: Page loaded in English

Course of test case

	External Event	Reaction
1	L'utilisateur sélectionne la langue	La librairie change la langue de l'application à la langue sélectionnée langue sélectionnée.

Success Criteria: Language changed to French

Question 3 (les tests sont aussi presents le repertoire devoir3 sur github)

Tout d'abord j'ai été capable de compiler et exécuter l'application en utilisant maven, le tout marchait à merveille comme le montre les screenshots suivant : (Avant d'ajouter mes propres tests)

```
D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp>java -jar ./target/BookstoreApp-0.1.0.jar
Press Enter to stop server
```



```
C:\Windows\System32\cmd.exe
[INFO] Total time: 02:47 min
[INFO] Finished at: 2021-07-20T14:27:38-04:00
[INFO]
D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp>mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] < SEG3103:BookstoreApp >
[INFO] Building BookstoreApp 0.1.0
[INFO] [ jar ]
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ BookstoreApp ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:compile (default-compile) @ BookstoreApp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.334 s
[INFO] Finished at: 2021-07-20T14:28:16-04:00
[INFO]
D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp>
```

```
C:\Windows\System32\cmd.exe
507
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul. 20, 2021 7:19:44 P.M. org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Starting ChromeDriver 91.0.4472.101 (af52a90bf87030dd1523486alcd3ae25c5d76c9b-refs/branch-heads/4472@(#1462)) on port 42
991
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul. 20, 2021 7:19:47 P.M. org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 7.265 s - in selenium.ExampleSeleniumTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.925 s
[INFO] Finished at: 2021-07-20T19:19:49-04:00
[INFO] -----

D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp>.
```

```
C:\Windows\System32\cmd.exe
[INFO] org/apache/ already added, skipping
[INFO] org/apache/commons/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/maven/org.apache.commons/ already added, skipping
[INFO] META-INF/LICENSE.txt already added, skipping
[INFO] META-INF/NOTICE.txt already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] org/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/LICENSE already added, skipping
[WARNING] Configuration options: 'appendAssemblyId' is set to false, and 'classifier' is missing.
Instead of attaching the assembly file: D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp\target\BookstoreApp-0
.1.0.jar, it will become the file for main project artifact.
NOTE: If multiple descriptors or descriptor-formats are provided for this project, the value of this file will be non-de
terministic!
[WARNING] Replacing pre-existing project main-artifact file: D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp\
target\BookstoreApp-0.1.0.jar
with assembly file: D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp\target\BookstoreApp-0.1.0.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:32 min
[INFO] Finished at: 2021-07-20T14:34:31-04:00
[INFO] -----

D:\kaana\SEG3503\lab1\seg3103_playground\lab6\BookstoreApp>.
```

Ensuite, j'ai pu développer les tests selon l'analyse faite dans les questions précédentes 1 et 2 voici les tests écrites dans Selenium à l'aide de WebDriver:

```
public void addBook(){
    driver.get("http://localhost:8080/admin");
    WebElement username = driver.findElement(By.id("loginId"));
    username.sendKeys("admin");
    WebElement password = driver.findElement(By.id("loginPasswd"));
    password.sendKeys("password");
    WebElement loginButton = driver.findElement(By.id("loginBtn"));
    loginButton.click();
    WebElement category = driver.findElement(By.id("addBook-category"));
    category.sendKeys("science");
    WebElement id = driver.findElement(By.id("addBook-id"));
    id.sendKeys("abcdef");
    WebElement title = driver.findElement(By.id("addBook-title"));
    title.sendKeys("title1");
    WebElement author = driver.findElement(By.id("addBook-authors"));
    author.sendKeys("author1");
    WebElement cost = driver.findElement(By.id("cost"));
    cost.sendKeys("10.5");
    WebElement addbtn = driver.findElement(By.name("addBook"));
    addbtn.click();
}

@Test
public void test1() {
    driver.get("http://localhost:8080/admin");
    WebElement username = driver.findElement(By.id("loginId"));
    username.sendKeys("admin");
    WebElement password = driver.findElement(By.id("loginPasswd"));
    password.sendKeys("password");
    WebElement loginButton = driver.findElement(By.id("loginBtn"));
    loginButton.click();
}
```

```

        WebElement category = driver.findElement(By.id("addBook-category"));
        category.sendKeys("science");
        WebElement id = driver.findElement(By.id("addBook-id"));
        id.sendKeys("abcdef");
        WebElement title = driver.findElement(By.id("addBook-title"));
        title.sendKeys("title1");
        WebElement author = driver.findElement(By.id("addBook-authors"));
        author.sendKeys("author1");
        WebElement cost = driver.findElement(By.id("cost"));
        cost.sendKeys("10.5");
        WebElement addbtn = driver.findElement(By.name("addBook"));
        addbtn.click();

        WebElement feedback = driver.findElement(By.id("feedback"));
        String actual = feedback.getText();
        String expectedResult = "Successfully added book";
        assertEquals(expectedResult, actual);
    }
}

```

```

@Test
public void test2() {
    driver.get("http://localhost:8080/admin");
    WebElement username = driver.findElement(By.id("loginId"));
    username.sendKeys("admin");
    WebElement password = driver.findElement(By.id("loginPasswd"));
    password.sendKeys("password");
    WebElement loginButton = driver.findElement(By.id("loginBtn"));
    loginButton.click();

    WebElement id = driver.findElement(By.id("addBook-id"));
    id.sendKeys("abcdef");
    WebElement title = driver.findElement(By.id("addBook-title"));
    title.sendKeys("title1");
    WebElement author = driver.findElement(By.id("addBook-authors"));
    author.sendKeys("author1");
    WebElement cost = driver.findElement(By.id("cost"));
    cost.sendKeys("10.5");
    WebElement addbtn = driver.findElement(By.name("addBook"));
    addbtn.click();

    WebElement feedback = driver.findElement(By.id("feedback"));
    String actual = feedback.getText();
    String expectedResult = "Validation errors\nA category is mandatory";
}

```

```

        assertEquals(expectedResult, actual);
    }

@Test
public void test3() {
    addBook();
    driver.get("http://localhost:8080");
    WebElement search = driver.findElement(By.id("search"));
    search.sendKeys("science");
    WebElement searchbtn = driver.findElement(By.id("searchBtn"));
    searchbtn.click();
    WebElement booktitle = driver.findElement(By.id("title-abcdef"));
    String expectedResult = "title1";
    String actual = booktitle.getText();
    assertEquals(expectedResult, actual);
}

@Test
public void test4() {
    driver.get("http://localhost:8080");
    WebElement searchbtn = driver.findElement(By.id("searchBtn"));
    searchbtn.click();
    WebElement addCart = driver.findElement(By.id("order-hall001"));
    addCart.click();

    WebElement cart = driver.findElement(By.id("cartLink"));
    cart.click();

    WebElement bookcost = driver.findElement(By.id("tothall001"));
    String expectedResult = "$39.95";
    String actual = bookcost.getText();
    assertEquals(expectedResult, actual);
}

@Test
public void test5() {
    driver.get("http://localhost:8080");
    WebElement searchbtn = driver.findElement(By.id("searchBtn"));
    searchbtn.click();
    WebElement addCart = driver.findElement(By.id("order-hall001"));
    addCart.click();

    WebElement cart = driver.findElement(By.id("cartLink"));
    cart.click();

```

```

WebElement number = driver.findElement(By.id("hall001"));
number.sendKeys("0");

WebElement update = driver.findElement(By.name("updateOrder"));
update.click();

WebElement bookcost = driver.findElement(By.id("tothall001"));

String expectedResult = "$399.50";
String actual = bookcost.getText();
assertEquals(expectedResult,actual);
}

@Test
public void test6() {
    driver.get("http://localhost:8080");
    WebElement searchbtn = driver.findElement(By.id("searchBtn"));
    searchbtn.click();
    WebElement addbook1 = driver.findElement(By.id("order-lewis001"));
    addbook1.click();
    WebElement addbook2 = driver.findElement(By.id("order-alexander001"));
    addbook2.click();

    WebElement cart = driver.findElement(By.id("cartLink"));
    cart.click();

    WebElement checkout = driver.findElement(By.name("checkout"));
    checkout.click();

    WebElement date = driver.findElement(By.id("order_date"));
    WebElement total = driver.findElement(By.id("order_total"));
    WebElement tax = driver.findElement(By.id("order_taxes"));
    WebElement shipping = driver.findElement(By.id("order_shipping"));

    String expectedResult = "2021/07/21$57.08$5.19$12.00";
    String actual = date.getText()+total.getText()+tax.getText()+shipping.ge
tText();
    assertEquals(expectedResult,actual);
}

@Test
public void test7() {
    driver.get("http://localhost:8080/admin");

```

```

WebElement username = driver.findElement(By.id("loginId"));
username.sendKeys("admin");
WebElement password = driver.findElement(By.id("loginPasswd"));
password.sendKeys("abcdef");
WebElement loginButton = driver.findElement(By.id("loginBtn"));
loginButton.click();

WebElement error = driver.findElement(By.className("content"));
String expectedResult = "Invalid username and/or password\nUser Name\nPa
ssword";
String actual = error.getText();
assertEquals(expectedResult,actual);
}

@Test
public void test8() {
    driver.get("http://localhost:8080/admin");
    WebElement username = driver.findElement(By.id("loginId"));
    username.sendKeys("admin");
    WebElement password = driver.findElement(By.id("loginPasswd"));
    password.sendKeys("password");
    WebElement loginButton = driver.findElement(By.id("loginBtn"));
    loginButton.click();

    WebElement logoutButton = driver.findElement(By.xpath("//input[@value='S
ign out']"));
    logoutButton.click();
    WebElement msg = driver.findElement(By.className("content"));

    String expectedResult = "You have been logged out\nUser Name\nPassword";
    String actual = msg.getText();
    assertEquals(expectedResult,actual);
}

@Test
public void test9() {
    addBook();
    WebElement addbtn = driver.findElement(By.name("addBook"));
    addbtn.click();

    WebElement feedback = driver.findElement(By.id("feedback"));

    String expectedResult = "Book with same id already exist";

```



```
String actual = feedback.getText();
assertEquals(expectedResult,actual);

}

@Test
public void test10() {
driver.get("http://localhost:8080");
Select language = new Select(driver.findElement(By.id("locales")));
language.selectByIndex(2);

String expectedResult = "http://localhost:8080/?lang=fr-CA";
String actual = driver.getCurrentUrl();
assertEquals(expectedResult,actual);
}
```

Et Ainsi tous les tests ont pu passer avec succès comme le demontre le screen suivant

EXPLORER

OPEN EDITORS

GROUP 1

Welcome

ExampleS... 2, U

GROUP 2

Java Test Report

BOOKSTOREAPP

bin

src

main

test\java

selenium

ExampleS... 2, U

ExampleTestj... U

target

bookstore5.jar U

pom.xml U

Welcome

ExampleSeleniumTest.java 2, U

src > test > java > selenium > ExampleSeleniumTest.java > ExampleSeleniumTest > test5()

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

@Test

Run Test | Debug Test | ✓

public void test3() {

addBook();

driver.get("http://localhost:8080");

WebElement search = driver.findElement(By.id("search"));

search.sendKeys("science");

WebElement searchbtn = driver.findElement(By.id("searchBtn"));

searchbtn.click();

WebElement booktitle = driver.findElement(By.id("title-abcdef"));

String expectedResult = "X";

String actual = booktitle.getText();

assertEquals(expectedResult,actual);

}

@Test

Run Test | Debug Test | ✓

public void test4() {

driver.get("http://localhost:8080");

WebElement searchbtn = driver.findElement(By.id("searchBtn"));

searchbtn.click();

WebElement addCart = driver.findElement(By.id("order-hall001"));

addCart.click();

WebElement cart = driver.findElement(By.id("cartLink"));

cart.click();

WebElement bookcost = driver.findElement(By.id("tothall001"));

String expectedResult = "\$39.95";

String actual = bookcost.getText();

assertEquals(expectedResult,actual);

}

Java Test Report

All 10 Passed 10

selenium.ExampleSeleniumTest

> test10

Passed

6.35s

> test1

Passed

3.54s

> test2

Passed

2.04s

> test3

Passed

3.82s

> test4

Passed

2.71s

> test5

Passed

2.18s

> test6

Passed

2.45s

> test7

Passed

2.64s

> test8

Passed

2.19s

> test9

Passed

2.6s