

Test Case

| TestCaseId | Component | Priority | Description/Test Summary | Pre-requisites |
|-----------------|--------------------|----------|--|--|
| Authorized_1 | Authorized_Pages | P1 | Verify that authorized pages can be viewed only by authenticated users. | User is logged in |
| Authorized_2 | Account_Management | P2 | Verify that authenticated users can change their account password. | User is logged in |
| Search_1 | Search_Movies | P0 | Verify that when a user searches for a movie by title, matching movie results are displayed | The system is online and available |
| BookTicket_1 | Booking_Module | P0 | Verify that a user can successfully book a ticket for a selected movie. | User is logged in |
| Refund_1 | Refund_Booking | P1 | Verify that a user can successfully request a refund for a previous booked ticket. | User is logged in and has a valid ticket booking. |
| Account_Login_1 | Account_Class | P0 | Verify that the login method correctly authenticates a user with valid credentials and rejects invalid ones. | An instance of the Account class with initialized attributed, including username and password. |

| | | | | |
|------------------|---|----|---|---|
| Payment_1 | Pay_Page | P0 | Verify that a correct payment method using Stripe or PayPal will result in a ticket purchase | User has money, has selected a movie to purchase a ticket for |
| Payment_2 | Pay_Page | P0 | Verify that a correct payment method using Stripe or PayPal will result in a ticket purchase | User does not have enough money, has selected a movie to purchase a ticket for |
| Showing_SetSeats | Showing_Class | P1 | Verify that setAvailableSeats method correctly updates the availableSeats attribute in the Showing class. | An instance of the Showing class with initialized attributes. A developer who can directly modify the object without purchasing a ticket or seat. |
| FullSystem_1 | Front_Page Login_Page Movie_Search_Page Movie_Showing_Selection_Page Ticket_Choice_Page Pay_Page Confirmation_Page Purchase_Managment_Page Account_Managment_Page | P0 | Verify that a user can navigate to all pages, and that they function together | Browser is launched |

| Template | | | |
|---|--|---|--------|
| Test Steps | Expected Result | Actual Result | Status |
| 1.Login to the website at https:// 2. Navigate to the Purchase Management Page and Account Management page in separate tabs. 3. Log out of the Movie Theater ticketing system and refresh both pages. | When logged in, you should have access to the protected pages. When not logged in, you should be given a not authorized error. | When logged in, the user is authorized to the protected pages. When not, the user is given an error. | Pass |
| 1. Write the url - https:// into the URL bar and press enter. 2. Navigate to the account management page. 3. Click on "Change Password" 4. Go to the email linked to the account, click on the password reset link. 5. Type in your new password in both text fields and click "Update Password" | When attempting to login, you should only be able to login with the new password you provided to the system. | When logging in, only the new credentials were valid. | Pass |
| 1. Launch the movie ticketing system. 2. Enter "Marvel" in the search bar and press enter. | Movies related to "Marvel" are displayed in the search results. | Movies such as "Avengers: Endgame," "Spider-Man: No Way Home," and "Black Panther" are displayed in the serch results. | Pass |
| 1. Select the movie "Fast and Furious 7" 2. Select the date, time, and seat. 3. Proceed to payment. 4. Confirm booking. | Ticket is booked successfully, and a confirmation message is displayed. | The booking fails with an error message: "Payment could not be processed." The user was charged, but no booking confirmation was generated. | Fail |
| 1. Go to the "My Bookings" section. 2. Select the ticket for "Fast and Furious 7" that needs a refund. 3. Click on "Requested Refund". 4. Confirm the refund request when prompted. | The refund request is submitted successfully, and a confirmation message is displayed indicating that the refund is being processed. | The booking fails with an error message: "Payment could not be processed." The user was charged, but no booking confirmation was generated. | Pass |
| 1. Create an instance of the Account class and set username to "testUser" and password to "securePassword". 2. Call the login(String password, String username) method with the correct credentials ("securePassword", "testUser"). 3. Assert that the method returns true for the correct credentials. 4. Call the login(String password, String username) method with incorrect credentials ("wrongPassword", "testUser"). 5. Assert that the method returns false for the incorrect credentials. | The login method should return true when provided the correct username and password, and false otherwise. | The correct credentials returned true, and any incorrect credentials return false. | Pass |

| | | | |
|--|--|--|------|
| 1. After confirming you wish to purchase a ticket on the Ticket Choice Page, click on either payment gateway. 2. Enter your corresponding payment information into the fields provided. 3. Click "Confirm Purchase" | After the payment is authorized, the ticket receipt appears on the screen. The dollar amount specified on the receipt has been withdrawn from the customer's payment method. | The receipt was shown after the payment was authorized, and the correct dollar amount was taken from the payment account. | Pass |
| 1. After confirming you wish to purchase a ticket on the Ticket Choice Page, click on either payment gateway. 2. Enter your corresponding payment information into the fields provided. 3. Click "Confirm Purchase" | The payment should be rejected within 3 seconds, and a notification notifying the user to "Try again, or use a different payment method." appears. | The payment was rejected, but does not notify the user within the proper time constraint of 3 seconds. | Fail |
| 1. Create an instance of the Showing class with an initial availableSeats value, e.g., 100. 2. Call the setAvailableSeats(int availableSeats) method with a new value, such as availableSeats - 1. 3. Retrieve the availableSeats attribute from the Showing instance. | The availableSeats attribute should accurately reflect the new value. | The field availableSeats was properly updated after calling the setAvailableSeats method. | Pass |
| 1. Write the url - https://().com in the browser's URL bar and press enter. 2. Once ().com is launched navigate to the login page 3. Enter username and password for test account and press enter then navigate back to front page 4. Navigate to the Movie Search Page and enter "Marvel" in the search bar and press enter 5. Select the first movie available taking the user to the Movie Showing Selection Page 6. Select the first available showing taking the user to the Ticket Choice Page 7. Enter a choice of 1 ticket and press enter taking the user to the Confirmation page 8. Select confirm taking the user to the Front Page 9. Navigate to the Purchase Management Page 10. Navigate back to the Front Page 11. Navigate to the Account Management Page 12. Navigate back to the Front Page | The Website opens up on the browser, Tester account is logged in, All pages are visited without crashing or error, Ticket is purchased | The Website opens up on the browser, Tester account is logged in, All pages are visited without crashing or error, Ticket is purchased | Pass |

| Test Executed By |
|------------------|
| James Johnson |
| James Johnson |
| Marvee Balyos |
| Marvee Balyos |
| Marvee Balyos |
| James Johnson |

James Johnson

James Johnson

James Johnson

Seth Blanchard