

Test Cases									
Test Case Id	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Executed By
Functional_Test_Case_01	User Account	High	Verify that users are able to log in if they have valid credentials	There must already be a user account created with known credentials	1. Navigate to login page 2. Enter a valid email and password 3. Click on the login button	User is logged in and directed to the homepage	User logs in and is able to access the website and it's homepage	Pass	Calvin
Functional_Test_Case_02	Ticket Booking	High	Ensure that users can book a ticket for a selected movie	User account logged in, movie available for booking	1. Log in to the user account. 2. Select a movie 3. Choose available seats. 4. Proceed to payment.	The selected seats are booked and a ticket is generated	User books seat and a ticket is generated	Pass	Calvin
Unit_Test_Case_01	Movie Class	High	Verify that the getFreeSeats() method returns only available seats	Movie object initialized with seat data	1. Initialize a Movie object with specific seat availability data. 2. Call getFreeSeats()	Method should return a map with only seats marked as available	Method returns a map with seats marked as available	Pass	Calvin
Unit_Test_Case_02	MakePaymentClass	Medium	Ensure confirmPurchase() generates a valid transactionID	MakePayment object initialized with Ticket	1. Create a MakePayment object. 2. Execute confirmPurchase() with a valid Ticket object.	A unique transactionID is generated and returned	Unique transactionID is generated and returned	Pass	Calvin
System_Test_Case_01	Movie Selection, User Account, Payment	High	Verify that a registered user can successfully book a movie ticket by selecting a movie, picking a time and	User must have an existing account and be logged in	1. Log in. 2. Go to movie selection page. 3. Select a movie, theater and time. 4. Pick seats and proceed to pay. 5. Pay with saved card information	User should be directed to the confirmation page with details of movie,theater and seat.	User is directed to the proper confirmation page with all details	Pass	Alex
System_Test_Case_02	User Account, Refund	Medium	Verify that a user can successfully request a refund and receive a free voucher ticket	User must have purchased ticket	1. Log in to user account. 2. Go to purchased tickets section. 3. Request a refund for the ticket. 4. Check account to see if voucher was issued	System processes refund and user account receives a free ticket vouched that can be redeemed	Refund voucher is given after processing and then seat is available again	Pass	Alex
Functional_Test_Case_03	Search Filter	High	Ensure that the search filter function display movie list and in the right order	User must have an existing account and be logged in A database of search results exists	1. Navigate the search bar 2. Select the filter function and apply filters 3. Confirm filter and press the search button	User is presented with a list of movies that match the filter in order of most matched to least matched	User is presented with a list of movies that match the filter in order of most matched to least matched	Pass	Minh
Functional_Test_Case_04	Search Bar Auto-correct	Medium	Verify that the search bar can handle incorrectly spelled search queries, and show the relevant movies.	User must have an existing account and be logged in A database of search results exists Spell-check/suggestions functionality is enabled	1. Navigate the search bar 2. Enter misspelled word in the search bar (e.g., "Openhemer" instead of "Oppenheimer" 3. Press the search button	Display a suggestion "Did you mean Oppenheimer?" and display results depending on the suggestion	Display a suggestion "Did you mean Oppenheimer?" and display results depending on the suggestion	Pass	Minh
System_Test_Case_03	Admin Management	Medium	Guarantees an admin can add, update, and delete movies, and that those changes are shown in user searches	Needs admin account logged in	1. Admin logs in 2. Add a new movie 3. Update movie showtimes 4. Delete movie 5. Verify changes on user side	User interface reflects all changes made by the admin	Website is successfully update by admin user	Pass	Justin
Unit_Test_Case_03	TheaterLccation	Low	Ensure listMovies() returns all movies added to the theater	TheaterLocation initialized with movies added to the theater	1. Create TheaterLocation object with multiple movie objects 2. Call listMovies()	Should return a list containing all added Movie objects	Returns list containing all added Movie objects	Pass	Justin