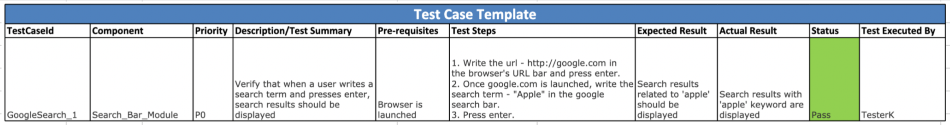
Test Case

TestCaseId	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Results	Status	Test Executed By
update_userInfo_1	user_database	P1	Ensure account info of user is updated after	User is logged in, app is open.	1. load user ticket info 2. Prompt to see user seat position 3. Check user's movie history	User's account history shows previous movie and seat position	User's account history shows previous movie and seat position	Pass	Tester Bob
promoCode_1	user_ticketinfo	P2	Ensure discount codes apply correctly	User has selected ticket and seat, then has navigated to checkout.	1. Enter a valid discount code at checkout. 2. Complete the purchase.	Discount is applied, and the total price is adjusted accordingly.	User applied promo code and the price was changed according to the discount.	Pass	Tester Sam
updateTicket_1	user_ticketinfo	P1	Update the user's ticket information corresponding to the theather	User has app opened and is purchasing ticket.	1. picks ticket 2. user goes to ticket checkout 3. ticket is purchased	User's ticket information is updated in the database	User will see their ticket in their movies tab	Pass	Tester Fred
theaterLocation_1	user_location theater_location	P3	Searches nearby theaters based on user's location	User has app opened and location enabled	1. User searches movies 2. Server prompts user's location 3. Provides list of theaters with requested movie nearby	User is prompted available theaters nearest their location	User is prompted available theaters nearest their location	Pass	Tester Matt
purchaseSnacks_1	theater_food	P3	Verify that the user can purchase snacks while purchasing tickets	User is in ticket checkout on app	1. User nears completion of ticket purchase 2. UI prompts food selection 3. User clicks food selection and purchases in tandem with ticket	User is able to purchase their selected snack along with their ticket	User has both ticket and snacks purchased	Pass	Tester Victor
searchMovie_1	movie_name	P1	Verify users can browse available shows.	User has navigated to the search for movie page	1. Log in to the application. 2. Access the browse shows section. 3. Select different filters (date, genre, venue).	User is able to select movie and the selection of movies changes due to filter changes	Movie selection changed as user applied filter, and the user was able to select a movie	Pass	Tester Xinping
chooseSeat_1	movie_name movie_seats	P1	Verify that only avaliable seats can be selected	User has chosen a movie and is on the seat selection page	1. Choose a show and access the seat map. 2. Select available seats. 3. Seat is confirmed	User is able to only select avaliable seats	User was able to select desired seats that was avaliable. Unavaliable seats were not selectedl	Pass	Tester Vincent
purchaseTicket_1	seats_database	P1	Test availability of seats in the theater	User searches movies for tickets	1.User prompts server for movies 2. Server checks availability of seats for selected movie 3. Server returns "full" if all seats are taken	User puchases ticket and selected seat is updated in database	Server updates seat availability and refreshes server	Pass	Tester Tran
sqlInjection_1	search_bar	P0	Ensure the application is resistant to SQL injection attacks.	Application is running as intended	1. Attempt to inject SQL code via input fields (search, login)	Application does not execute the injected SQL code.	Nothing happens due to the SQL code not being executed	Pass	Tester Nguyen
performanceUnderload_1	ticket_app	P0	Verify the application's performance under high user load.	Have several devices accesing the ticketing app. The app previously works as intended	1. Simulate multiple users accessing the application simultaneously. 2. Perform various actions (browse shows, purchase tickets).	Application remains responsive with no significant delays.	The application continues to function as normal	Pass	Tester Richard



Test Case Template

TestCaseId	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Executed By
GoogleSearch_1	Search_Bar_Module	P0	Verify that when a user writes a search term and presses enter, search results should be displayed	Browser is launched	1. Write the url - http://google.com in the browser's URL bar and press enter. 2. Once google.com is launched, write the search term - "Apple" in the google search bar. 3. Press enter.	Search results related to 'apple' should be displayed	Search results with 'apple' keyword are displayed	Pass	TesterK