

Test Case Template									
TestCaseId	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Executed By
UnitTest_1	a function or method	P1	Viewing movie showtimes	Website is launched, movie is	1. Clicking on a theatre location 2. Clicking on a movie title	All movie showtimes related to movieTitle should be returned	Movie Showtimes associated with the movie title was successfully returned	Pass	RogerD
FunctionalTest_1	User Registration	P2	Customer regristration (does the regristration actually register a user)	Customer is on the page for regristration through manual entry, or clicking yes after entry of "Login" during purchasing process.	1. Entry of username, psasword, email, phone number 2. Submit of entry 3. Verification of username, password, email, phoneNumber	New account information should be added to the user database	New account matching the entered information is found in the database	Pass	RogerD
SystemTest_1	User Signin	P2	Customer sign-in	Customer account is already registered	1. Entry of username and password 2. Submission of username & password 3. Verification of username & password 4. Display of success/failure page	If the username and password is verified against the user database then a successful login page is returned. Otherwise an error message is returned.	Account found and login credentials are valid, successful login screen is shown.	Pass	RogerD
FunctionalTest_2	Ticket Submission	P1	Verify that when a user submits a ticket the customer service team receives it.	Ticket form is filled out	1. Navigate to help section of website. 2. Fill out ticket information 3. Submit ticket	Customer Support Ticket database should update with new ticket and accurate information	Customer Support Ticket database has the new ticket containing the filled out information	Pass	CameronL
UnitTest_2	Payment class	P0	Verify that the Payment class is able to request and receive payment	Ticket is selected	1. Tickets selected for purchase 2. Select payment method 3. Payment information entered 4. Wait for module to verify payment method of choice	If payment is approved, tickets should be removed from availability and an email with the tickets should be sent	Tickets removed from purchasable selection and email is sent to the recipient with the tickets	Pass	CameronL
SystemTest_2	movie, showtime, Theater. Ticket	P0	Tests the overall process of selecting seats and purchasing tickets from start to finish	Browser is launched	1. Navigate to website 2. Search for movie 3. Select a showtime 4. Select available seats 5. Checkout tickets	Process should be intuitive and easy, seat selection should be accurate and up to date, and tickets should be received once payment is processed.	Website navigation is smooth and works well. Tickets were able to be purchased.	Pass	CameronL
UnitTest_3	Reserveticket method	P0	Tests to see if the seating map is updated when a ticket is selected. Will check to see if seating is returned if seat is not purchased.	Ticket is reserved (not purchased)	1. Select time and seating 2. No not purchase ticket 3. Check in another browser for the same time and seat 4. Wait 10 minutes 5. Refresh second browser to see if the seat is now available	The creation of a ticket should update the seating map proerply and mark it unavailable for another customer. If not purchased within 10 minutes, the seat is	The seating map is accurate and up to date with the system. Unbought tickets are returned and purchase tickets move the seat avaiability even after 10	Pass	MatthewD
UnitTest_4	Customer Class invalid handling	P0	Checks to see that only valid inputs areaccepted when seleceng seat amout.	Customer ticket is being created. Website is asking for user input.	1. Search for time and seating 2. When prompted for amount of tickets, enter number more than available seats. 3. Wait for rejection message	Before the specific seat is chosen, the user must input the amount of tickets they're purchasing. If the input exceeds availbilty, Customer will see message saying "Not	Proper error messages were sent given invalid amount of tickets.	Pass	MatthewD

UnitTest_5	Customer Class duplicate ticket prevention	P0	Test to ensure that duplicate tickets arent purchased at the same time.	Two users select identical tickets are the exact same time.	1. Set up multiple bots with the same speed to execute ticket purchase 2. Both tickets being purchased are the sametime 3. Only one of the two bots will be allowed, the other will be rejected	Using a Queue system, requestes are handled one at at time. If working, only one of the multiple requests will be allowed.	Only one bot was able to purchase a ticket, the rest were denied and directed back to select another ticket.	Pass	MatthewD
UnitTest6	Theatre class duplicate information prevention	P0	Verify that there are no duplicate times, location, and information from available theatres. Reject any duplicates from the system.	Search result should only show one available time, per movie, per theatre.	1. Proxy a duplicate theatre and pass it into the database 2. Search for time and movie 3. Purchase ticket	If the same identification for a thatre's data is duplicate, only one will be displayed and the others are deleted. All input and requests on the website will return to the	One of each time and theatre was present regardless of duplicated data. Purchasing a ticket updated properly.	Pass	MatthewD