The Solidity contract file is an NFT contract named NFTicketmaster based on the Ethereum blockchain. It imports the ERC721 standard and IERC721Receiver interface from OpenZeppelin, which is commonly used for NFTs (Non-Fungible Tokens). Given this context, here's a basic outline for a test plan for the NFTicketmaster contract:

1. Set Up and Initialization
    a. Ensure the testing environment is properly configured, with tools and frameworks in place;
    b. Deploy the NFTicketmaster contract and verify successful deployment;
    c. Confirm initial values of variables, such as ‘owner’, ‘totalOccasions’, ‘totalSupply’;
2. Functional Test Cases
    a. Owner Functionality:
        i. Validate that the deployer is set as the owner;
        ii. Test owner-only functions like ‘list’ and ‘withdraw’ to ensure they reject access from non-owner accounts.
    b. Minting NFTs
        i. Mint tickets for valid occasions, and check updates in 'totalSupply,'' 'hasBought,'' 'seatTaken,’ and' 'seatsTaken.''
        ii. Test edge cases: invalid occasion ID, seat already taken, insufficient payment, exceeding maxTickets limit.
    c. Listing New Occassions
        i. Testing listing occasions with valid parameters, and verifying correct data storage.
    d. Ticket Purchase
        i. Transferring NFTs is the operational scenario for purchasing and transferring tickets.
    e. ERC721 Compliance and Receiver Implementation
        i. Ensure compliance with ERC721 standards and proper receiver implementation.
3. Operational scenarios
    a. Use-case scenarios: such as a user buying multiple tickets, selling a ticket on the secondary market, or an event being fully booked.
    b. Pre-conditions: listing for users has enough balance.
    c. Post-conditions: correct ticket assignment and funds transfer.
    d. Edge cases: max ticket limit, purchasing post-limit, buying for past or non-existent occasions.
4. Security testing
    a. Test critical functions like ‘ticketpurchase’ and ‘withdraw’ for reentrancy vulnerabilities;
    b. Ensure that functions restricted to the owner or ticket owners behave as expected.
5. Performance and gas usage
    a. Assess gas usage across critical operations like minting, transferring, and purchasing.
    b. Identify potential optimizations for cost-effective.

Test Scenario: User Buying Multiple Tickets

To validate that the NFTicketmaster contract correctly handles a single user purchasing multiple tickets for an event.

Pre-conditions

The NFTicketmaster contract is deployed and operational.
An event (occasion) is listed in the contract with a sufficient number of tickets available.
The user (buyer) has enough Ethereum balance to purchase multiple tickets.

Test Steps:

1. List an Occasion:

As the owner, list a new occasion with a specified number of tickets.

2. Multiple Ticket Purchase:

As a user, initiate the purchase of multiple tickets in separate transactions.
Optionally, test the purchase of multiple tickets in a single transaction if the contract supports it.

3. Validate Ticket Allocation:

Verify that the correct number of tickets is minted and assigned to the user.
Ensure the seatTaken, hasBought, and seatsTaken mappings are updated accurately.

4. Check Ticket Limit:

Attempt to buy more tickets than available for the occasion and expect the transaction to fail.

5. Verify Event Data:

Check that the occasion's data (like the number of tickets sold) is updated correctly.

6. Test Funds Transfer:

Ensure the correct amount of funds is transferred from the buyer’s account to the contract.

7. Repeat with Different Users:

Repeat the purchase process with different user accounts to ensure consistent behavior.

Post-conditions

The total number of tickets sold for the occasion is updated.
Each ticket is uniquely assigned to the user.
The user’s Ethereum balance is reduced by the correct purchase amount.
The contract’s balance is increased by the total sale amount.

Edge Cases to Test

Purchasing the maximum number of tickets available.
Trying to purchase tickets after the maximum limit for the occasion has been reached.

Purchasing tickets for a non-existent or past occasion.