Verification Test Plan

TEST CASES:

searchbar_1:

- Scope:
  - The user has entered valid characters into the search bar.
  - The user already has logged in through guest or account login.
- Overview:
  - This is a unit test case for the usage of the search bar module of the system and is how the user will search for their wanted movies.
- Ideal Test Result:
  - The user clicks the search bar text box, inputs a string and presses enter.
  - The user inputs alphanumeric characters that are related to movies and not just nonsense
  - The system will then present the user with a catalog of movies either containing their term, close to the term, or related.
- Real Test Result:
  - The user has logged in as a guest or an account and has clicked on the search bar module.
  - The user then inputs “action” and presses enter
  - The user is presented with a catalog of movies containing the word action or has been tagged with action.
- Conclusion/Recommendation:
  - The user is able to access the search bar module and is able to search using keywords, terms, or names to locate their wanted movie.

login_1:

- Scope:
  - User has entered valid login info.
- Overview:
  - This is a unit test case, where the user has logged into the website and has accessed the account login feature.
- Ideal Test Result:
  - The user clicks on the account login and is able to login:
  - The user will click the username text box and that box is able to accept alphabets and numbers. The user will enter their username and then move on.
  - The user then clicks the password text box which also accepts alphabets and numbers. The user will then input their password into the password text box and click login.

- Real Test Result:
    - The user clicks the account login and inputs their info:
    - The user will click the username text box and input “user123”
    - The user will click the password text box and input “pass123”
    - The user is then logged in
- Conclusion/Recommendation:
    - A user with a registered account is able to login with their credentials and access the rest of the movie ticketing website.

login_2:

- Scope:
    - User has accessed the website and is capable of navigation to the module.
- Overview:
    - This is a functional test where the user has accessed the website and wants to browse for movies without a registered account.
- Ideal Test Result:
    - The user will access the website then click the guest login text box instead of account login.
    - The user will be logged into a guest account and be allowed to access the rest of the website and browse movies.
- Real Test Result:
    - The user has accessed the website and clicks guest login, the guest login works and the user is then allowed to access the rest of the website.
- Conclusion/Recommendation:
    - The guest login functionality works and allows a user that does not want to create an account or login to an account to access and use the website to purchase tickets for their wanted movie.

profile_1:

- Scope:
    - User has logged in with valid info.
    - Users are able to navigate to their profile and booking history tab.
- Overview:
    - This is a functional test case for the usage of the booking history module of the account profile module and allows a user to see their booking history.
- Ideal Test Result:
    - The user has logged into their account with the correct information and wants to access their purchase history.
    - The user then clicks on their profile and selects the booking history tab and is redirected to a page where they can see their booking history.

- Real Test Result:
    - The user will login to their account with a valid username and password. The user will then click their profile and are presented with a drop down, they then click booking history in that drop down menu and are redirected to a page where they can see their booking history.
- Conclusion/Recommendation:
    - The booking history module works correctly and allows for a logged in user to access their booking history through their profile and see purchases made.

auth_token:

- Scope:
    - Valid account credential
- Overview:
    - This tests the functionality and the validity of authenticating a valid user. This test involves secure communication between client server and Account Management Server (AMS). If the test passes, users will be able to access their account management via the website.
- Ideal Test Result:
    - Assuming the user inputs valid account credentials to the login page, the data is transmitted directly to the AMS to verify if the provided account credential is legit and if it is currently active. If both of the conditions are met, AMS generates an authentication token and its token is securely encrypted using AES algorithm. Once encrypted, its encrypted authentication token will be transmitted back to the client user via secure network protocol HTTPS and certificates like SSL/TLS to ensure data is secure when being transmitted. Once the authentication token is received by the client server, it uses the authentication token to validate and verify the provided account credential is legit and safe. This is indicated by granting account access upon submitting a valid account credential.
- Real Test Result:
    - Assuming the user inputs valid account credentials to the login page, the user was redirected to the homepage as a logged in user. The logged in user now has access to managing its own settings. They have the ability to manage coupons, change password, change email address, change payment method, change username, etc.
- Conclusion/Recommendation:
    - Overall, the test was successful. The Expected Test Result matched the Real Test Result. Account credential data and authentication tokens were securely transmitted between the client server and AMS. One recommendation to implement stronger security is to utilize the upcoming quantum computing algorithm to encrypt account credential data and authentication tokens. By doing

so, it will be practically impossible to crack its data, if ever it was sniffed by intruders.

transaction_req1:

- Overview:
    - This tests the secure data transmission from and to client server and designated bank server. All actions are recorded in the Centralized Log Server (CLS) for the system to ensure appropriate evidence for future purposes.
- Ideal Test Result:
    - Assuming the user inputs the right bank information, its data should be transmitted directly to the bank server through passing AWS Sage Maker. The ideal test result should track only action records in the CLS and not the actual transaction data. To ensure user privacy, CLS should only keep records of actions, not data of the transactions.
- Real Test Result:
    - Assuming the user inputs the right bank information, its data were successfully encrypted throughout the whole process of transaction. CLS also only recorded its actions, not its data. Transactions were successful and transaction data was only visible by the bank and the client server. No transaction data was left and recorded by our system to ensure user privacy.
- Conclusion/Recommendation:
    - Overall, the test was successful. The Expected Test Result matched the Real Test Result. CLS successfully recorded only actions of the transaction and not its data. One recommendation to implement stronger security is to ensure the AWS Sage Maker server is super secure since that is the only server in our system that is able to plainly view the bank data.

transaction_req1:

- Overview:
    - This tests the functionality of AWS Sage Maker and how accurate it detects potential fraud transaction requests. If the AWS Sage Maker flags the transaction request, it should take appropriate actions to handle this error.
- Ideal Test Result:
    - Assuming the user inputs a flagged fraud bank information, its encrypted data should go to AWS Sage Maker and it should flag if the bank information is a potential fraud. Once flagged, the client server should be prompted with the following: “Invalid input, try again”.
- Real Test Result:
    - Assuming the user inputs a flagged fraud bank information, its encrypted data was flagged by the AWS Sage Maker and the client server was prompted with “Invalid

input, try again". The AWS Sage Maker successfully detected a potential fraud transaction request. It only flags it, it does not take necessary actions. A custom built script inside the AWS Sage Maker automates detected frauds to prompt the error message back to the client server.

- Conclusion/Recommendation:
    - Overall, the test was successful. The Expected Test Result matched the Real Test Result. AWS Sage Maker was able to successfully detect potential fraud transactions. By doing so, it saved the corporate tons of money that might have gone to the criminal's hand. One recommendation to implement a stronger security is to add another server in between the client server and the AWS Sage Maker to increase security layers. By doing so, it increases the strength of server infrastructure security.

seat_selection_1:

- Scope:
    - User has selected a valid seat
- Overview:
    - This test aims to evaluate the functionality of the seat diagram and ensure that users can successfully select available seats. If the test passes, users should be able to finalize their ticket purchase along with their selected seat.
- Ideal Test Result:
    - The seat diagram displays all of the seats in the theater clearly, including the layout of the theater.
    - Users can easily select seats by clicking or tapping on them.
    - Selected seats are visually highlighted or marked.
    - Users can proceed to finalize their ticket purchase after selecting a seat.
- Real Test Result:
    - The seat diagram displays all of the seats in the theater clearly, including the layout of the theater.
    - Users can easily select seats by clicking or tapping on them.
    - Selected seats are visually highlighted or marked.
    - Users select seats that are already taken by another user.
    - Users can proceed to finalize their ticket purchase after selecting a seat.
- Conclusion/Recommendation:
    - The seat selection functionality of the module performs as expected. Users can effectively select available seats without encountering any issues. It is recommended to proceed with further testing to ensure consistent performance across different browsers and devices. Additionally, user feedback should probably be gathered to identify any potential usability improvements to the seat selection process.

seat_selection2:

- Scope:
    - User has selected an invalid seat
- Overview:
    - This tests the system's response when a user attempts to select an invalid seat that is already taken by another user. The expected behavior is that the user receives a prompt informing them that the selected seat is unavailable, and they are returned to the seat selection diagram to choose another seat.
- Ideal Test Result:
    - User attempts to select an invalid seat.
    - System prompts the user with a message indicating that the seat is already taken.
    - User is redirected back to the seat selection diagram.
    - Users can choose another seat.
- Real Test Result:
    - User attempts to select an invalid seat.
    - System displays a message informing the user that the seat is already taken.
    - User is redirected back to the seat selection diagram.
    - Users can choose another seat.
- Conclusion/Recommendation:
    - The test has successfully verified that the system handles the scenario of selecting an invalid seat appropriately. Users are informed about the unavailability of the selected seat and are directed back to the seat selection diagram to choose an alternate seat.

auth_token2:

- Scope:
    - Authentication token is not granted to customers with invalid account credentials
- Overview:
    - This tests the system's behavior when a user attempts to log in with invalid credentials. The test verifies that the user is not granted an authentication token if the login information is incorrect. Additionally, it checks if the system allows the user three login attempts with invalid credentials before locking the account. If the account is locked, the system should prompt the user to perform two-factor verification using their registered phone number or email.
- Ideal Test Result:
    - User enters invalid login credentials.
    - System denies authentication tokens and notifies the user that they entered invalid information.
    - The user is given three more attempts to log in with incorrect credentials.
    - If the user enters correct credentials after < 3 attempts, the user will be logged in.
    - After three failed attempts, the user's account is locked.

    - System prompts the user to perform two-factor verification using their registered phone number or email.
- Real Test Result:
    - User enters invalid login credentials.
    - System denies authentication tokens and notifies the user that they entered invalid information.
    - The user is given three more attempts to log in with incorrect credentials.
    - If the user enters correct credentials after < 3 attempts, the user will be logged in.
    - After three failed attempts, the user's account is locked.
    - System prompts the user to perform two-factor verification using their registered phone number or email.
- Conclusion/Recommendation:
    - Auth_token2 passes the test. The system accurately handles invalid login attempts by denying authentication tokens and implementing security measures such as two-factor verification. It is recommended to conduct further testing to ensure that the lockout mechanism functions as intended under various scenarios and to make the user experience of the two-factor verification process easier.