Test Plan

This section outlines the test plans for the verification and validation of the Movie Theatre Ticketing System (MTTS). The test plan is structured to ensure that the system performs as intended, and all requirements are fulfilled. Each test targets specific features of the design, with tests classified by **granularity**: unit tests, functional tests, and system tests. The design diagram has been modified to show the scope and targets of the tests, and specific failure scenarios are included where relevant.

Test Objectives

The objective of this test plan is to ensure that the MTTS system:

- Accurately processes user actions (e.g., login, movie selection, payment).
- Effectively communicates with the backend to retrieve and update data.
- Integrates and operates correctly across all components, including the **Rewards System** and **Email Notification System**.
- Handles error cases and edge cases effectively without breaking the user experience.

Scope of the Test Plan

The test plan covers:

- **Unit Tests**: Individual components like managers and databases will be tested to ensure their behavior in isolation.
- **Functional Tests**: Specific features like login, seat selection, and payment processing will be tested to verify they work as described in the specifications.
- **System Tests**: The system as a whole will be tested, including end-to-end booking processes, rewards application, and admin report generation.

Test Cases

Each test case has been designed to target a specific feature of the system. The test cases are divided across the three levels of granularity:

Unit Testing

- **Target Component**: **User Manager**
  - **Test Case**: Validate that the User Manager correctly checks correctly handles user authentication (login/logout)
  - **Test Vector**: Input correct/incorrect credentials and observe system behavior.
  - **Targeted Failures**: Failure to authenticate users or incorrect handling of login errors.
  - **Sample**: **TS_MTS_001 –** Test Login Functionality

- **Target Component**: **Seat Manager**
  - **Test Case**: Validate the seat availability status and update when a user selects a seat.
  - **Test Vector**: Simulate seat selection and verify that the seat is reserved.
  - **Targeted Failures**: Failure to update seat status.
  - **Sample**: **TS_MTS_004 –** Test Seat Selection Functionality

Functional Testing

- **Target Feature**: **Movie Selection**
  - **Test Case**: Verify that users can browse and filter available movies.
  - **Test Vector**: Select a genre and verify that the correct movies are displayed.
  - **Targeted Failures**: Incorrect or missing movie listings.
  - **Sample**: **TS_MTS_005 –** Test Browse Movies Functionality
- **Target Component**: **Payment Processor**
  - **Test Case**: Ensure the Payment Processor applies rewards and processes payment correctly.
  - **Test Vector**: Test with and without rewards applied, and ensure payment amounts are updated accordingly
  - **Targeted Failures**: Failure to apply rewards or process payments.
  - **Sample**: **TS_MTS_002 –** Test Payment Processing
- **Target Component**: **Food and Drink Add-ons**
  - **Test Case**: Verify users can add and modify food and drink options during checkout.
  - **Test Vector**: Select and modify add-ons, then proceed to payment.
  - **Targeted Failures**: Failure to update order summary with correct add-ons.
  - **Sample**: **TS_MTS_006 –** Test Food and Drink Add-on Functionality
- **Target Component**: **User Registration**
  - **Test Case**: Verify successful user registration, ensuring user information is correctly saved and a confirmation email is sent.
  - **Test Vector**: Test the registration process by inputting valid account information, submitting it, and observing the system's response.
  - **Targeted Failures**: Failure to create an account, database errors, or missing confirmation email.

    - **Sample**: **TS_MTS_003 - Test User Registration**

System Testing

- **Target Feature**: **Complete Booking Process**
    - **Test Case**: Simulate a user logging in, selecting a movie, applying rewards, completing payment, and receiving a confirmation email.
    - **Test Vector**: Test the full booking process from login to confirmation email.
    - **Targeted Failures**: Errors in booking, payment, or confirmation.
    - **Sample**: **TS_MTS_007 –** Test Booking Process
- **Target Feature**: **Admin Report Generator**
    - **Test Case**: Verify that the admin can generate sales reports based on booking data.
    - **Test Vector**: Input a wide date range and generate a report.
    - **Targeted Failures**: Failure to generate a complete and accurate report.
    - **Sample**: **TS_MTS_008** - Test Generation Report for Admin
- **Target Feature**: **Conflict Seat Selection**
    - **Test Case**: Simulate two users selecting the same seat at the same time and verify conflict handling.
    - **Test Vector**: Simulate both users proceeding to payment and see how the system handles the conflict.
    - **Targeted Failures**: Incorrect seat availability or double-booking.
    - **Sample**: **TS_MTS_009** - Test Conflict Seat Selection
- **Target Feature**: **Cancel Booking and Refund**
    - **Test Case**: Verify users can cancel a booking and receive the correct refund.
    - **Test Vector**: Simulate booking cancellation and check if the refund is processed.
    - **Targeted Failures**: Failure to process refunds or incorrect refund amounts.
    - **Sample**: **TS_MTS_010** - Test Cancel Booking and Refund Process

Test Coverage and Strategy

Test Sets/Vectors:

- **Granularity**: Each test is designed to address a specific granularity, unit, functional, or system, covering a range of inputs and outputs for each feature.
- **Failure Cases**: The test plan addresses a variety of potential failures, such as invalid inputs (e.g., wrong login credentials), system errors (e.g., database connection issues), and edge cases (e.g., concurrent seat selections).

Test Plan Execution:

- Each test case will be executed in the development environment and will include a detailed report of actual results compared to expected results.
- Testing will be repeated in the production environment to ensure the transition from development to production does not introduce bugs or errors.

Validation and Verification Strategy:

- **Verification**: Ensures the system meets the functional and technical requirements specified in the design.
- **Validation**: Ensures the system fulfills the end-user needs and performs as intended in real-world scenarios.

6. Conclusion

This test plan ensures comprehensive testing of the MTTS, covering all key features and components. By identifying failure cases and outlining specific test vectors, we aim to validate the functionality of the system across all user scenarios, ensuring a robust and reliable user experience.