MoviePlug Ticketing System

Verification Test Plan

Version 1

October 19, 2023

Group 5

Ruben Guillen

Justin Pudiquet

Rosa Lisa Silipino

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2023

1. Introduction

The purpose of this Verification Test Plan is to provide an in depth overview of functional, unit and system test plans that satisfy business requirements.

2. Test Plan Scope

The verification tests will target the following aspects of the software system:

- Unit Testing - Individual class methods and functions.
- Functional Testing - The interaction between different classes and functions.
- System Testing - The end-to-end behavior and performance of the software system.

3. Test Levels

3.1. Unit Testing

Unit tests analyze and evaluate at a basic level of testing individual components of MoviePlug Ticketing System software. These components can include objects and classes within the system.

3.2. Functional Testing

Functional testing is a test level that focuses on verifying the functional aspects and features of the software application. The primary objective of functional testing is to ensure that the software behaves as expected, meets its functional requirements, and delivers the intended functionality to end-users.

3.3. System Testing

System testing focuses on assessing the system's behavior and performance as a whole rather than individual components. System testing verifies that all the components work together seamlessly and meet the system's non-functional requirements, such as performance, security, and scalability.

4. Test Cases

4.1. Unit Testing

4.11. Test Case 1: Booking

a) Description: Verify ticket booking timeout

b) Test Input: User will select seat(s) for movie but delay payment for an extended period.

c) Expected Output: Booking selection times out and selected seats are released.

d) Pass/Fail Criteria:

   i) Pass if booking times out

   ii) Fail if otherwise

4.12. Test Case 2: Verify System's Ticket Limit for Ticket Purchases

a) Description: This test will verify that the seating() class function communicates with the event management system (EMS) API to ensure conditions are set to limit the number of premium seats selected on the user's interface. The maximum number of seats a user can reserve is 20 seats utilizing MoviePlug. Upon finalizing movie, theater and showtime selection, the user will be navigated to the seat selection page if a premium theater is chosen. A graphical 2D layout of the theater allows users to select the specific seat(s) they chose. After exceeding the maximum number of seats (20) a popup error message should alert the user that the ability to choose any more seats is unable to be processed. This should mitigate surplus unfair ticket purchases. Possible faults that could occur is that the condition isn't properly set within the external API or that seating() class doesn't pull seating information from the database.

b) Test Input:

   i) Navigate to the seating selection page.

   ii) Select 21 seats on the 2D-seating map.

   iii) Click the "Next" button to navigate to the ticket selection page.

c) Expected Output:User receives an error message indicating the maximum number of seats have been selected.

d) Pass/Fail Criteria:

   Pass:

   i) System alerts number of seats selected exceed limit.

   ii) Booking() function

   Fail:

   i) System allows users to select over 20 seats.

4.13. Test Case 3: Verify Senior and Military Discounts

a) Description: The purpose of this test is to verify that Users who have been identified as either a Senior Citizen or a Military Service member or an affiliate receive a discount on the purchase of movie tickets regardless of the movie, theater, or screening selected. This test case verifies that the Payment() class receives the valid information containing the discount and that it is properly applied to the transaction.

b) Test Input:

i) Go through the process of selecting a movie, screening, theater location, seat (if applicable).

ii) Just prior to confirming purchase look at the Cart Summary there should be an inline credit named "Discount" in parentheses for "$xx.xx" dollar amount.

iii) After submitting payment the proper amount is charged to the payment method.

c) Expected Output: Cart Summary shows "x" amount of tickets added to cart minus "$xx.xx" dollar amount for Senior/Military Discount. After purchasing, the amount reflects ticket price minus appropriate discount.

d) Pass/Fail Criteria:

i) Pass - The test passes if the total amount charged reflects the amount of a ticket or multiple tickets minus the amount of the discount. (i.e. if ticket is $15, and Military/Senior Discount is 10% then the amount charged per ticket for a customer who is discount eligible would be $13.50/ticket)

ii) Fail - This test will fail in any event where the total amount charged is anything other than $13.50/ticket for users who are discount eligible.

4.2. Functional Testing

4.21. Test Case 1: Search Functionality

a) Description: Verify user can search for movies

b) Test Input: Movie title

c) Expected Output: Movie matching the search criteria is displayed

d) Pass/Fail Criteria:

i) Pass if correct movie is displayed

ii) Fail if otherwise

4.22. Test Case 2: Verify Users Can Reset Password Using Forgot Password

a) Description: This black box test will verify functionality of User() class function, integration with Email API, database updates as well as server communications. Main focus is to ensure the user(s) can properly reset passwords when either locked out or the password is no longer known. Forgot Password button is found on the login page, once clicked it will trigger password recovery/reset transactional email sent by API. User(s) should receive an email containing a link that will navigate them to the SSL secure password reset page and will be prompted to input a new password. Upon successful input of a valid password, the new password should be updated within MoviePlug's internal database and authentication server and work during the next login. Possible faults could occur at the internal database with failure to update the user's password information. Process is dependent on the Email API to send email to the user for the reset process to occur.

b) Test Input:
   i) Navigate to the login page.
   ii) Click the "Forgot Password" button.
   iii) Enter valid email associated with account.
   iv) Click the "Reset Password" button.
   v) Receive password reset email.
   vi) Redirect to password reset page.
   vii) Enter a valid new password and verify.
   viii) Click "Confirm" Button.
   ix) Return to the login page and try the new password.

c) Expected Output: User enters a valid email associated with an account to get sent a password reset link to secure SSL encrypted page to reset their password. Upon valid input of a new password that follows password requirements and user confirmation to reset password, the password associated with the account will be changed for login credentials.

d) Pass/Fail Criteria:

Pass:
   a. Password meets security criteria and is accepted by authentication.
   b. Database updates and saves the new password.

Fail:
   a. User() class failed to retrieve new password from database.
   b. The internal database didn't store the updated password.
   c. Email API didn't successfully initiate transactional email.

4.23. Test Case 3: Verify the ticket(s) purchased contain the proper theater, movie, seat information selected.

a) Description: The purpose of this test is is to ensure that the Booking(), Ticket(), Seat(), Theater(), Movie(), and Screening() classes all function accordingly and that they all pass the proper information over to the Ticket() class which will print out the adequate data as selected and paid for by the user.

b) Test Input:
   i) Search for a movie.
   ii) Select a theater.
   iii) Select a screening time.
   iv) Select number of desired tickets.
   v) If applicable, select a seat(s).
   vi) Confirm cart & proceed to finalize purchase.
   vii) Confirmation page and ticket(s) will populate and prompt you to print/save.

viii) Confirm that ticket data concurs with all options chosen by the customer.

c) Expected Output: Printed/Saved ticket will contain all information as chosen by customer (i.e. customer chose to buy 3 tickets for movie "A" in theater "X" and seats "E8,E9,E10", printed tickets will reflect that information

d) Pass/Fail Criteria:

i) Pass - This test case passes if the tickets printed properly reflect the users order (i.e. if User selected movie “X”, at screening time “xx:xx”, in a premium theater “T” and select seat “A1”. Then the ticket should reflect the user name with movie “X” in theater “T” allocated to seat “A1”.

ii) Fail - This test case fails if any component of the ticket does not match the user’s order (i.e. incorrect movie selection, screening time, theater location or seat selection *(if applicable)*).

4.24. Test Case 4: Verify that the movies are being shown as "available" apply to those under 10 minutes past screening time or at a later date.

a) Description: The purpose of this test is to ensure that the system will check multiple classes throughout the system like the Movie(), and Screening() class to make sure that the system will allow a user to purchase a ticket for a movie that has not been playing for longer than 10 minutes past the screening time.

b) Test Input:

i) Search for a movie.

ii) Select a theater. (note that some locations will be available while others are grayed out meaning they are not available)

iii) Select a screening time, for those theaters that are marked as available.

c) Expected Output: Movie "A" will be available to be seen at theater "X" assuming the movie hasn't been rolling for more than 10 minutes or if it is for any screening at a later time.

d) Pass/Fail Criteria:

i) Pass - This test case passes if the application allows the user to purchase a ticket for a screening that is either at a later time or if it has not been more than 10 minutes past the screening time. (i.e. movie starts at 4pm, system should allow user to purchase ticket until 4:10pm)

ii) Fail - This test case fails if the application allows the user to purchase a ticket for a screening that is more than 10 minutes past the screening time.

4.3. System Testing

4.3.1. Test Case 1: User Authentication

a) Description: Verify that the user can log into their system.
b) Test Input: User log in unique credentials: Username and Password
c) Expected Output: User login successfully
d) Pass/Fail Criteria:
  i) Pass if user credentials are correct
  ii) Fail if otherwise

4.3.2. Test Case 2: User Authentication

a) Description: Verify invalid log in
b) Test Input:
  i) Username invalid and password invalid
  ii) Username correct and password invalid
  iii) Username invalid and password correct
c) Expected Output: User receives an error message indicating invalid credentials
d) Pass/Fail Criteria:
  i) Pass if error message displays
  ii) Fail if otherwise

4.3.3. Test Case 3: Verify Confirmation Email After Transaction is Processed

a) **Description**:
b) **Test Input**:
  1. Navigate to the payment page.
  2. Pay with preferred payment method.
  3. Confirm payment information
  4. Click "Checkout" Button
c) **Expected Output**: User receives on screen confirmation with text stating confirmation email with tickets information has been sent to email address on file. Upon checking the user checking their personal email inbox, corresponding email from MoviePlug will be delivered confirming purchase details and NFT ticket access.
d) **Pass/Fail Criteria**:
  i) Transactions must be processed.
  ii) User must receive a confirmation email.

5. Test Case Matrix

Please refer to SRC for Test Case Matrix