# SDS: Test Plan

Group 2

Justin Pelak, Lucas Fredricks, Andy Boyd, Gilbert Venegas

**Basic Layout**

**Introduction:**

This document aims to ensure the correct functionality of the TrustyTicket system. The scope outlines what is and is not being tested. Give an outline of the processes and steps taken to ensure a stable system build. Give an outline of methodology used when developing the test cases.

**Purpose:**

The purpose of the TrustyTicket system test plan is to: - Verify core functionalities of our ticketing system, ensuring that it behaves as expected across all our various scenarios. - Verify features such as user login, ticket purchases, movie availability checks, and refund requests to confirm.

**General User Scope of Testing**

**In Scope**

- Login functionality
    - Tests if the login page sends the correct error messages for incorrect user credentials
    - Expected to throw an error at the user, such as “Incorrect Username or Password.”
    - Additional case of locking account after too many failed consecutive login attempts
- Movie Theater Page
    - Tests if page correctly displays movies as sold out or available accordingly
    - Expected to keep users from visiting the page of sold out movies and purchasing sold out movie tickets
- User Profile Editing
    - Tests if user information can be updated and is updated in reasonable time
    - Expected to update user information and display profile changes in real time
- User registration and sign in functionality
    - Tests if a new user is able to create an account once sent to the sign up page
    - Tests if a newly registered user is able to sign in
    - Expected to create new user account and redirect new user to login page
    - Expected to allow user to login with newly created login credentials
- Refund request system functionality
    - Tests if user can request a refund for a purchased ticket
    - Expected to send confirmation message and email to user and corresponding email, stating they successfully requested a refund
    - Expected to send a fail message to user if refund request was unable to be sent

**Out of Scope**

- Payment Functionality
- Movie Theater database
- Ticket Sending/printing

**Test Schedule**

1) UserSignUp_ID00004; Sign_Up, New_User, Theater_User_Data, User_Sign_In
2) TestCase_ID00001; Login_Bar_Module
3) UserSignIn_ID00005; User_Sign_In, Theater_User_Data, Movie_Theater_Page
4) CheckAvailability_ID00006; Movie_Theater_Page, Check_Availability, Choose_Another_Movie
5) TestCase_ID00002 ; Movie_Theater_Page
6) TestCase_ID00003; Profile_Editing
7) TestCase_ID00007; User_Registration
8) TestCase_ID00008; Refund_Request
9) TestCase_ID00009; Ticket_Purchase_Validation
10) TestCase_ID00010; Ticket_Purchase_Page

**Testing approaches**

- Login functionality
    - First, enter incorrect login credentials; alternate between incorrect username and incorrect password
    - In both cases, the exact same error message is displayed to attempt to circumvent malicious users and brute force attacks
    - If first step passes, continue to enter incorrect credentials; the system should automatically lock down the user account for some reasonable amount of time
    - If this case also passes, the test can be concluded as a pass
- Movie Theater Page
    - First, create default “sold out” movie theater for testing
    - User view should display the movie with a gray “sold out” bar where purchase should usually be
    - If this step passes, attempt to click on the sold out bar; the button should not link to anything
    - If this case also passes, the test can be concluded as a pass
- User Profile Editing
    - First, log in to a test user account and visit the “Edit User Information” page
    - User should be able to change various parts of their account; ensure that username, password, profile picture, age, email, phone number, and payment details are able to be changed/removed accordingly
    - If this step passes, attempt to reload the page and ensure updated details have been saved to the database
    - If this case also passes, the test can be concluded as a pass
- User registration and sign in functionality
    - First, log out of any current user instances and navigate to the default log in screen
    - Select “Register new account”; user should be redirected to a registration page
    - If this step passes, enter an already registered username and then an invalid password
    - The system should send an error message stating that the username is taken, or password is invalid (too short, not enough special characters, etc.) respectively

        - If this step passes, enter a unique username and valid password; the system should send a confirmation message and redirect you to the login page
        - Log in with the newly created credentials
        - If this case also passes, the test can be concluded as a pass
- Refund request system functionality
    - First, log in to a test user account and visit the “My Tickets” page
    - User view should display a list of tickets, with those that are upcoming sooner than others first, and the past, now unavailable tickets at the bottom
    - If this step passes, navigate to an invalid, expired ticket and try to request a refund; the button should not exist for this case
    - If this step passes, navigate to a valid ticket and try to request a refund; after a series of confirmations, prompt the user to enter their password and if they enter valid credentials, send the user a confirmation message and email that their refund will be sent according to the payment gateways’ details
    - If this case also passes, the test can be concluded as a pass

**Methodology**

- Unit Testing
    - Verify the integrity of code in individual functions / components. Thus ensuring proper functionality at a small scale.
    - Expected errors: Syntax, boundary.
- Functionality testing
    - Verify functionality of individual features within the system from an end user perspective ensuring compatibility between components.
    - Expected errors: Incorrect output, network, validation, integration, compatibility.
- System Testing
    - Verify system meets SRS requirements as well as works as a cohesive unit for the end user.
    - Expected errors: SRS requirement not met, data flow, database/ data integrity errors, configuration, integration.

**Test Cases**

https://github.com/BoostedJ/CS250/blob/main/SDS/SDSTestCases.pdf (PDF)
https://github.com/BoostedJ/CS250/blob/main/SDS/SDSTestCases.xlsx (Spreadsheet)

**Risks and Issues**

- No connection with any databases
- Too many orders being processed
- Sign in not validating
- System double books movie seat
- Refunds not sent to users
- Profile editing not updated to database
- Payment gateway under maintenance during testing
- Registering new user with taken username overrides previous user