SDS: Test Plan

For The Movie Theater Ticketing System

By: Nathan Gozun, Adam Larson, Anthony Barbaro, Kellen Murphy, Kobe Tran

Version: 1.0

Publish: March 22, 2023

Design Specification:

The updated design specification includes three classes: ShoppingCart, Checkout, and Account Management.

[Shopping Cart] is responsible for keeping track of the user's selected tickets and their total cost. It has the following attributes:

Cart: an array of Ticket objects
totalCost: an integer that represents the total cost of the tickets in the cart
discount: a boolean value that indicates whether a discount has been applied
It also has the following methods:

addTicket(Ticket tik): adds the specified Ticket object to the cart
removeTicket(Ticket tik): removes the specified Ticket object from the cart
displayCart(): returns an array of Ticket objects that represents the current contents of the cart
addDiscount(int amount): applies a discount of the specified amount to the total cost of the tickets in the cart
[Checkout] is responsible for handling the user's purchase once they have finished selecting their tickets. It has the following attributes:

cart: a ShoppingCart object that represents the user's selected tickets
totalCost: an integer that represents the total cost of the user's purchase
It also has the following methods:

ConfirmPurchase(): confirms the user's purchase and updates the appropriate records in the system
checkout(): handles the payment process and completes the purchase
[Account Management] is responsible for managing user accounts. It has the following attributes:

Username: a String that represents the user's username
Password: a String that represents the user's password
It also has the following methods:

createnewAccount(): creates a new user account with the specified username and password
deleteAccount(): deletes the user's account
changePassword(String newPW): changes the user's password to the specified new password.

TestCaseID	Component	Priority	Description/Test Summary	Pre-requisites
Ticketing_Ticket	Ticket Class	P1	Verify that the ticket object is created with the correct movie title and time	A movie must exist to assign the ticket to
Ticketing_Movie	Movie Class	P1	Verify that the movie object is created with the correct rating	None
Ticketing_Payment1	Payment System Class	P2	Verify that the payment system correctly calculates the total price when there is no discount	None
Ticketing_Payment2	Payment System Class	P2	Verify that the payment system correctly calculates the total price when there is a discount	None
Ticketing_Schedule1	Master Schedule Class	P1	Verify that a movie can be added to the master schedule	None
Ticketing_Schedule2	MasterSchedule Class	P1	Verify that a movie can be removed from the master schedule	None
Ticketing_Review_Class	Review Class	P1	Verify that a review can be created with a rating and description	None
Ticketing_Shopping_Cart	Shopping Cart	P1	Verify that a customer can add a product to the shopping cart and the product is displayed in the cart.	Customer is logged in and is viewing a product page.

Ticketing_Checkout_Class	Checkout Class	P2	Verify that a customer can complete a checkout process successfully. Customer has added items to the shopping cart.	User is signed in and product is already in checkout cart
Ticketing_Account_Management	Account Management	P1	erify that a user can change their password and the new password is saved in the system.	User is logged in and is on the 'Change Password' page.

Test Steps	Expected Result	Actual Result	Status	Test Executed by
1. Create a new Ticket object with the movie title "The Matrix" and time "9:00 PM". 2. Retrieve the values of the movie title and time from the Ticket object.	The movie title should be "The Matrix" and the time should be "9:00 PM".	The results shown was "The Matrix" at "9:00 PM"	Passed	Kellen Murphy
1. Create a new Movie object with the rating "PG-13". 2. Retrieve the value of the rating from the Movie object.	The rating should be "PG-13".	The results shown was that the movie rating is "PG-13".	Passed	Kellen Murphy
1. Create a new Payment System object with a ticket price of $10. 2. Call the totalPrice() method on the Payment System object.	The total price should be $10.	Result shown was total price being $10	Passed	Adam Larson
1. Create a new Payment System object with a ticket price of $10 and a discount amount of $2. 2. Call the totalPrice() method on the Payment System object.	The total price should be $8.	Result shown was total price being $8	Passed	Adam Larson
1. Create a new MasterSchedule object. 2. Create a new Movie object. 3. Call the addMovie() method on the MasterSchedule object with the Movie object as a parameter. 4. Call the getMovies() method on the MasterSchedule object.	The list of movies should contain the newly added movie.	Newly added movie was added to the list of movies	Passed	Kobe Tran
1. Create a new MasterSchedule object. 2. Create a new Movie object. 3. Call the addMovie() method on the MasterSchedule object with the Movie object as a parameter. 4. Call the cancelMovie() method on the MasterSchedule object with the Movie object as a parameter. 5. Call the getMovies() method on the MasterSchedule object.	The list of movies should not contain the removed movie.	The removed movie was not in the list of movies	Passed	Kobe Tran
1. Create a new Review object with a rating of 4 and a description of "Great movie!". 2. Retrieve the values of the rating and description from the Review object.	The rating should be 4 and the description should be "Great movie!".	The results shown were that movie was rated a 4 with a description Great movie!"	Passed	Anthony Barbaro
1. Click on the 'Add to Cart' button on the product page. 2. Verify that the product is displayed in the shopping cart.	The product should be added to the shopping cart and displayed to the customer.	The results shown were that the product was added and displayed to shopping cart	Passed	Anthony Barbaro

1. Navigate to the checkout page. 2. Fill in the shipping and billing information. 3. Select a payment method. 4. Confirm the order.	The order should be completed successfully and the customer should receive a confirmation email.	The results shown were that the order was completed and confirmation was sent to the user's email	Passed	Nathan Gozun
1. Enter the current password and the new password in the respective fields. 2. Click the 'Save' button. 3. Verify that the new password is saved in the system.	The new password should be saved in the system and the user should be able to log in using the new password.	New password was saved to system	Passed	Nathan Gozun

Verification Test Plan: Ticketing System

Unit Testing:

Test Set 1: Ticket Component Constructor
Test Vector 1.1: Create a new Ticket object with valid inputs for movie, time, location, and age.
Test Vector 1.2: Create a new Ticket object with invalid inputs (e.g. blank movie name, negative time value, invalid location, etc.).
Test Vector 1.3: Verify that the Ticket object is created with the correct attribute values.

Test Set 2: Payment System Calculate Total Price Method
Test Vector 2.1: Calculate the total price for a single-ticket purchase with no discounts applied.
Test Vector 2.2: Calculate the total price for a group ticket purchase with a discount applied.
Test Vector 2.3: Verify that the calculated price is correct and matches the expected result.

Functional Testing:

Test Set 3: Ticket Purchase Functionality
Test Vector 3.1: Select a movie from the Master Schedule and purchase a single ticket with a valid payment method.
Test Vector 3.2: Attempt to purchase a ticket for a movie that is sold out or unavailable.
Test Vector 3.3: Verify that the purchased ticket is added to the user's account and that the Master Schedule is updated with the correct availability information.

Test Set 4: Cancel Ticket Functionality
Test Vector 4.1: Cancel a ticket purchase before the movie start-time and receive a full refund.
Test Vector 4.2: Attempt to cancel a ticket purchase after the movie start-time and receive an error message.
Test Vector 4.3: Verify that the canceled ticket is removed from the user's account and that the Master Schedule is updated with the correct availability information.

System Testing:

Test Set 5: Registration and Login Functionality
Test Vector 5.1: Create a new registered account and verify that the user's personal and payment information is stored correctly.
Test Vector 5.2: Attempt to login with invalid credentials and receive an error message.
Test Vector 5.3: Verify that the user can access their account and view their purchase history and account details.

Test Set 6: Integration Testing with External Payment Systems
Test Vector 6.1: Test the Ticketing System with a valid credit card payment method.
Test Vector 6.2: Test the Ticketing System with a valid PayPal payment method.
Test Vector 6.3: Verify that the Payment System correctly processes the transaction and updates the user's account information.

These test sets target various features of the Ticketing System, including ticket creation, payment processing, cancellation, user registration and login, and integration with external payment systems. By testing at the unit, functional, and system levels, we can ensure that the system is working as expected and handling various use cases and scenarios.