# Homework 3 starts on page 27

# Software Design Specification

Version 2

October 19th, 2023

Group #3

Jake Stonebraker
Jose Urrutia
Emre Yikilmaz

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

# **Revision History**

Date	Description	Author	Comments
October	Version 1	Group 3	<first revision=""></first>
5th, 2023			
October	Version 2	Group 3	<second revision=""></second>
19, 2023			<added plan="" test="" verification=""></added>

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
		Software Eng.	
	Dr. Gus Hanna	Instructor, CS 250	

# **Table of Contents**

Revision History	2
Document Approval	2
1. Introduction	
2. System Overview	4
3. System Architecture	5
3.1 UML Class Diagram	
3.2 Individual Classes of System	5
3.2.1 Theater	5
3.2.2 Person	6
3.2.3 Account	6
3.2.4 Admin	7
3.2.5 Customer	8
3.2.6 Employee	8
3.2.7 Showtime	9
3.2.8 Movie	10
3.2.9 Seat	11
3.2.10 Reservation	12
3.2.11 Ticket	14
3.2.12 TicketType	16
3.2.13 Payment	17
3.2.14 Receipt	18
3.3 Software Architecture	
3.3.1 Software Architecture Diagram	19
3.3.2 Software Architecture Description	20
4. Development Plan and Timeline and Partitioning of Task	s24
4.1 Timeline	
4.1.1 Phase 1: Project Initiation	25
4.1.2 Phase 2: Design Planning	25
4.1.3 Phase 3: Development	25
4.1.4 Phase 4: Deployment	26
4.1.5 Phase 5: Launch	26
5. Verification Test Plan	27
5.1 Introduction	27
5.1.1 Purpose	27
5.1.2 Scope	27
5.1.3 Objectives	27
5.2 Test Objectives	27
5.2.1 Unit Testing	27
5.2.2 Functional Testing	27

5.2.3 Performance Testing	28
5.2.4 Security Testing	28
5.2.5 Compatibility Testing	28
5.3 Testing Approach	28
5.3.1 Unit Testing	28
5.3.1.1 Seat Reservation Functionality	28
5.3.1.2 Calculating the Total Price Functionality	29
5.3.1.3 Find Reservation Functionality	29
5.3.2 Functional Testing	30
5.3.2.1 Booking a Reservation	30
5.3.2.2 Filtering Showtimes	31
5.3.3 Performance Testing	31
5.3.3.1 Multiple Users Try Booking at the Same Time	31
5.3.3.2 System Integration With Payment Gateway	32

# 1. Introduction

The introduction of the Software Design Specification (SDS) provides an overview of the SDS with purpose, product description, scope, and the objective of this document. The aim of this software design specification is to describe the software organization and development plan.

# 1.1 Purpose

The purpose of the document is to define the design for the overall software architecture and implementation of the theater ticketing system. The classes, data structures, together with all major functions and their parameters will be organized and specified. It should help the developers to adequately implement the software components as specified in this document.

# 1.2 Scope

This software design specification is to be used by our Software Engineers as a template definition of the design and architecture that is to be used in implementing the Theater Ticketing System.

#### 1.3 Objective

The Theater Ticketing System is meant to provide the customer and theater employee the ability to view, select, reserve, purchase, and refund tickets. Customer accounts will also be created (if customer wishes to do so) to remember personal information, preferences, and payment methods.

# 2. System Overview

# 2.1 Product Perspective

The product enables users (customer/employee) to view, sort, select, reserve, purchase, and refund tickets for theaters the web browser and purchase tickets online. Employees of the cinema can see and manage bookings from ticket buyers. Besides that, cinema administrators can manage user accounts. Using the online ticketing system the ticket purchasing process becomes easier and more efficient for both ticket buyers and the cinemas. Enhanced features like account registration enable opportunities for loyalty programs and discounts for ticket buyers.

#### 2.2 Product Functions

- 2.2.1 <u>Theater class functions</u>: displayTheaterDetails()
- 2.2.2 Person class functions: n/a
- 2.2.3 <u>Account class functions</u>: resetPassword(), validate\_Username()
- 2.2.4 <u>Admin class functions</u>: addMovie(), addShowtime(), deleteUser()
- 2.2.5 **Customer class functions**: getReservation(), initiateRefund()
- 2.2.6 <u>Employee class functions</u>: searchReservation(), initiateRefund()
- 2.2.7 <u>Showtime class functions</u>: displayShowtime()
- 2.2.8 <u>Movie class functions</u>: sortMovies(), getMovieRating()
- 2.2.9 <u>Seat class functions</u>: isOccupied(), occupy(), vacate()
- 2.2.10 <u>Reservation class functions</u>: calculatePrice(), addSeat(), removeSeat(), timeLimit(), cancelReservation(), createReservation()
- 2.2.11 <u>Ticket class functions</u>: getUser(), getShowTime(), getSeatInfo(), checkTicketType()
- 2.2.12 **Ticket type functions:** verifyDiscount(), setPrice()
- 2.2.13 **Payment class functions**: wipeCardInfo(), processPayment(), refund()
- 2.2.14 **Receipt class functions**: generateReceipt()

#### 2.3 User Characteristics

The intended user is any customer looking to reserve and purchase a ticket online or at a kiosk at any of the theater locations. Employees will be able to make reservations and purchases on customer's behalf when necessary.

#### 2.4 General Constraints

This application will only run on a system that supports a GNU C++ compiler. Is meant to be run on the three most popular web browsers: Google Chrome, Firefox, Safari.

# 2.5 Assumptions and Dependencies

The user will have basic knowledge of how to navigate a website using a mainstream web browser, have a valid email address, locate their desired theater, select the movie of interest, enter personal and payment information, and present their ticket at the movie theater.

# 3. System Architecture

# 3.1 UML Class Diagram

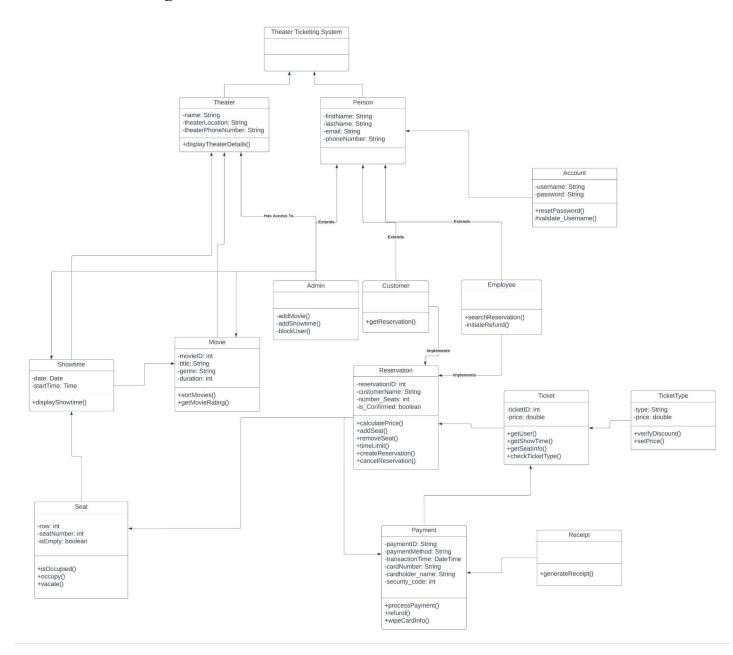


Figure 1: UML Class Diagram containing the class objects and their relationships

# 3.2 Individual Classes of System

# 3.2.1 Theater

the Theater class is will be the object class that can create instances of the individual theaters within the theater ticketing system

#### 3.2.1.1 Attributes of Theater class

#### **3.2.1.1.1** theaterName

the name of the theater will be held in data type: String

#### 3.2.1.1.2 theaterLocation

the address of the theater will be held in data type: String

#### 3.2.1.1.3 theaterPhoneNumber

the phone number of the specific theater will be held in data type: String

# 3.2.1.2 Functions available in Theater class

# 3.2.1.2.1 displayTheaterDetails()

this function will serve to display the attributes of the specific theater

#### **Arguments:**

this function will contain no arguments as it will directly handle the attributes within the class

# **Return Type:**

this function will have a void return type as it will simply display the details of the theater in a neat manner

#### 3.2.2 Person

the Person class will be the object class that will create instances of each individual actor within the Theater Ticketing System

#### 3.2.2.1 Attributes of Person class

#### **3.2.2.1.1** firstName

the first name of the person will be held in data type: String

#### 3.2.2.1.2 lastName

the last name of the person will be held in data type: String

#### 3.2.2.1.3 email

the email of the person will be held in data type: String

# **3.2.2.1.3 phoneNumber**

the phone number of the person will be held in data type: String

### 3.2.3 Account

the Account class will have an associative relationship with the Person class as each instance of the person class will have an account corresponding to the type of "Person"

#### 3.2.3.1 Attributes of Account class

#### **3.2.3.1.1** username

the username of the instance of the person class will be held in data type: String

#### **3.2.3.2.1** password

the password of the instance of the person class will be held in data type: String

#### 3.2.3.2 Functions available in Account class

#### 3.2.3.2.1 resetPassword()

this function will allow the instance of the person class (whether a customer, admin, or employee) to reset their password when they have forgotten it or wish to change it for security reasons

# **Arguments:**

-new\_password (str): the new password for the user

# **Return Type:**

-boolean: Will return true if the password reset is successful, false if not successful

## 3.2.3.2.2 validate\_Username()

this function will check whether or not the username given by the user is taken by another user

#### **Arguments:**

there are no arguments required for this function as it will directly deal with the username attribute of this class

# **Return Type:**

-boolean; will return true if username is available, false if username is not available

### **3.2.4 Admin**

the Admin class will be the specific object class that creates instances of each individual that has administrator privileges in the Theater Ticketing System. They will have access to the Showtime and Movie classes in order to add movies and showtimes for each Theater object

#### 3.2.4.1 Attributes of Admin class

the admin class will inherit the attributes of the Person class

#### 3.2.4.2 Functions available in Admin class

#### 3.2.4.2.1 addMovie()

this function will allow the admin to create the instance of a Movie Object that will be added to the list of movies for the specific Theater object they want to add it to

# **Arguments:**

this function has no arguments as the admin class will have access to the movie class and its attributes

# **Return Type:**

-void: the function will return nothing

# **3.2.4.2.2** addShowtime()

this function will allow the admin to create the instance of a Showtime Object that will be added to the list of showtimes for the specific Movie object they want to add it to

### **Arguments:**

this function has no arguments as the admin class will have access to the Showtime class and its attributes

# **Return Type:**

-void: the function will return nothing

# **3.2.4.2.3 deleteUser()**

this function will allow the admin to delete Customer Class instances in the person database

## **Arguments:**

this function has no arguments as the admin class will have access to the person class and its attributes

# **Return Type:**

-void: the function will return nothing

#### 3.2.5 Customer

the Customer class will be the specific Object class that will create instances of each individual wanting to purchase tickets using the Theater Ticketing System

#### 3.2.5.1 Attributes of Customer class

the Customer class will inherit the attributes of the Person class

#### 3.2.5.2 Functions available in Customer class

## 3.2.5.2.1 getReservation()

this function will allow the Customer to search for their existing reservation

# **Arguments:**

-reservationID (int): This function will use the reservation ID to use in a search algorithm in the Theater Ticketing System database

# **Return Type:**

-boolean: the function will return true if the reservation is found or false if it is not found.

# 3.2.6 Employee

# 3.2.6.1 Attributes of Employee class

the Customer class will inherit the attributes of the Person class

# 3.2.6.2 Functions available in Employee class

## 3.2.6.2.1 searchReservation()

this function will allow the Employee to search for a customer's reservation in order to print tickets or initiate a refund.

# **Arguments:**

-reservationID (int): This function will use the reservation ID to use in a search algorithm in the Theater Ticketing System database

# **Return Type:**

-boolean: the function will return true if the reservation is found or false if it is not found.

#### 3.2.6.2.2 initiateRefund()

this function will allow the Employee to initiate a refund to the Customer if a reservation is found and it abides by the refund policy as described in the functional requirements. If the argument captured in the parameters is true, the function will initiate the refund

# **Arguments:**

-reservationFound (bool): This function will use the reservation ID to use in a search algorithm in the Theater Ticketing System database and capture the boolean from the searchReservation() function within the same Employee Class

#### **Return Type:**

-void: the function will return nothing.

# 3.2.7 Showtime

an instance of the Showtime class will be created by the administrator to be added under the movie database for the specific instance of the Theater class. It will contain the date and start time for the specific movie. It will also have access to the attributes of the Movie class that will be displayed to the user

# 3.2.7.1 Attributes of Showtime class

#### 3.2.7.1.1 date

the date will contain the date of the instance of the Showtime object using data type: Date

#### **3.2.7.1.2** startTime

the startTime will be initialized with the data type: Time

#### 3.2.7.2 Functions available in Showtime class

#### 3.2.7.2.1 displayShowtime()

this function will allow the Employee to initiate a refund to the Customer if a reservation is found and it abides by the refund policy as described in the functional requirements. If the argument captured in the parameters is true, the function will initiate the refund

# **Arguments:**

-reservationFound (bool): This function will use the reservation ID to use in a search algorithm in the Theater Ticketing System database and capture the boolean from the searchReservation() function within the same Employee Class

# **Return Type:**

-void: the function will return nothing.

# **3.2.8** Movie

the Movie class will hold the information necessary for the primary actor to view and search for the specific movie they want to see

#### 3.2.8.1 Attributes of Movie class

#### 3.2.8.1.1 movieID

each movie instance will have a unique ID that will have a data type: int

#### 3.2.8.1.2 title

each movie instance will have a title that will use the data type: String

#### 3.2.8.1.3 genre

each movie instance will have a genre classifier that will use the data type: String

#### 3.2.8.1.4 duration

each movie instance will have a run time that will be in minutes using the data type: int

#### 3.2.8.2 Functions available in Movie class

#### **3.2.8.2.1 sortMovies()**

this function will allow the user to filter movies based on genre or movie rating

# **Arguments:**

-this function has no parameters as the logic inside the function will determine whether the movie will be filtered by genre or movie rating, which will call the getMovieRating() function within the same class.

# **Return Type:**

-void: the function will return nothing.

#### 3.2.8.2.2 getMovieRating()

this function will interact with a third-party website such as IMDB or Rotten Tomatoes to get information on a specific movie's rating

# **Arguments:**

-title (Movie Object ): This function will use the title from an instance of a Movie object to search for the movie rating from the third-party website

# **Return Type:**

-the function will return the rating of the movie which will be displayed with a data type: String

# 3.2.9 Seat

the Seat class is responsible for creating an instance for each individual seat for a specific Showtime instance. It will interact with the Reservation class for when either a customer or employee wants create a reservation for a specific movie

#### 3.2.9.1 Attributes of Seat class

#### 3.2.9.1.1 row

the row for the specific seat will be initialized with the data type: int

#### **3.2.9.1.2** seatNumber

the actual seat number will be initialized with the data type: int

#### 3.2.9.1.3 is Empty

this attribute will flag whether or not the seat is empty with data type: boolean

#### 3.2.9.2 Functions available in Seat class

## **3.2.9.2.1 isOccupied()**

this function will interact with the seating chart interface and will display the seat in black if it is already purchased or red if it is still available to purchase

#### **Arguments:**

-this function has no parameters since it will interact directly with the attributes within the class

# **Return Type:**

-boolean: the function will return true if the seat is occupied or false if the seat is empty

# 3.2.9.2.2 occupy()

this function will interact with the seating chart interface to mark that is empty as occupied after a reservation is confirmed with payment confirmation

# **Arguments:**

- -isOccupied (bool): This function will call the isOccupied() function within the class to see if the seat is empty and ready for purchase.
- -reservationConfirmed(bool): this function will also call the reservationConfirmed() function to make sure that payment was processed before marking the seat as occupied

# **Return Type:**

-boolean: the function will return true if occupied successfully and false, if otherwise

# 3.2.9.2.3 vacate()

this function will interact with the seating chart interface and will turn an occupied seat back to red (previously black) after a successful reservation cancellation

#### **Arguments:**

-this function has no arguments as it will be called within the cancelReservation() method within the Reservation class

#### **Return Type:**

-void: the function will return nothing.

# 3.2.10 Reservation

the Reservation class will hold the attributes and functions of each single unique reservation instance created by either the Customer or the Employee. It will include a unique numerical ID, the name of the customer, and the number of seats they purchased

#### 3.2.10.1 Attributes of Reservation class

#### **3.2.10.1.1** reservationID

the reservation ID will be unique to the customer using a random number generator of type: int

#### **3.2.10.1.2** customerName

the name of the Purchaser will be initialized by user input with data type: String

#### **3.2.10.1.3** is Confirmed

this will be a flag for when the reservation is confirmed: type boolean

#### **3.2.10.1.4** number seats

this will represent the number of seats the user intends to purchase with data type: int

#### 3.2.10.2 Functions available in Reservation class

#### **3.2.10.2.1** calculatePrice()

this function will calculate the total price of the reservation based on the number of seats chosen for each Ticket Type. It will also include the sales tax and include a reservation fee calculated in the total

#### **Arguments:**

-this function will have no parameters as it will interact directly with the variables within the class itself.

#### **Return Type:**

-double: the function will a floating point double containing the total price of the reservation

#### 3.2.10.2.2 addSeat()

this function will see what Ticket Type the user wishes to purchase and will increment number\_seats variable by one after each time the program takes user input of a mouse click on a plus sign on the website interface

#### **Arguments:**

-this function will take a user input of a single mouse click on the plus sign located on the website interface

# **Return Type:**

-void: the function will return nothing.

#### **3.2.10.2.3** removeSeat()

this function will see what Ticket Type the user wishes to purchase and will decrement number\_seats variable by one after each time the program takes user input of a mouse click on a minus sign on the website interface

# **Arguments:**

-this function will take a user input of a single mouse click on the minus sign located on the website interface

# **Return Type:**

-void: the function will return nothing.

# **3.2.10.2.4** timeLimit()

this function will start a 10 minute timer once the user chooses to create a reservation. It will kick them out of the reservation window once time runs out

#### **Arguments:**

-This function contains no parameters

# **Return Type:**

-void: the function will return nothing.

#### 3.2.10.2.5 cancelReservation()

this function will allow the user to cancel a reservation by providing their unique reservation ID. The function will search the reservation database and will return a boolean notifying the user whether or not their reservation was found. The function will interact with the refund method to initiate the refund process

#### **Arguments:**

-reservationID (int): This function will use the reservation ID to use in a search algorithm in the Theater Ticketing System database.

#### **Return Type:**

-boolean: the function will return true if reservation is found and refund is processed

# 3.2.10.2.6 createReservation()

this function will initiate the process of creating a new reservation prompted by either the Customer or the Employee. It will interact with the Seat Class for them to be able to choose a Movie and Showtime. Once chosen, the function will interact with the functions within the Reservation class to add or remove seats and to calculate the total price. Once total price is calculated, the Payment class will be called where the processPayment() function will be invoked to process the payment.

# **Arguments:**

-this function takes no arguments

# **Return Type:**

void: the function will return nothing. It will change the is\_Confirmed variable to true which will trigger the system to prompt the user if they want an email or print receipt

# **3.2.11 Ticket**

The instance of a Ticket class will be generated once a reservation has been made in the Reservation class. The amount of instances created will depend on the number of seats purchased.

#### 3.2.11.1 Attributes of Ticket class

#### 3.2.11.1.1 ticketID

Each ticket will have a unique ID which has the data type: int

#### 3.2.11.1.2 price

the price of each ticket displayed will be determined by the Ticket Type. This variable will have the data type: double

#### 3.2.11.2 Functions available in Ticket class

#### 3.2.11.2.1 getUser()

this function will retrieve the customer name from the Customer class to be displayed on the ticket

#### **Arguments:**

-this function has no arguments

#### **Return Type:**

-void: the function will return nothing.

#### **3.2.11.2.2 getShowTime()**

this function will retrieve the Showtime information from the Showtime class to be displayed on the ticket

# **Arguments:**

-this function has no arguments

# **Return Type:**

-void: the function will return nothing.

# **3.2.11.2.3** getSeatInfo()

this function will retrieve the seat information from the Seat class to be displayed on the ticket

# **Arguments:**

-this function has no arguments

# **Return Type:**

-void: the function will return nothing.

# 3.2.11.2.4 checkTicketType()

this function will verify the ticket type: adult, child, student, or senior. This will invoke the verifyDiscount() method in the TicketType subclass. If the ticket type is verified, it will apply the discount

# **Arguments:**

This function has no arguments

# **Return Type:**

-boolean: the function will return true if the ticket type is a discounted type and verified correctly

# **3.2.12 TicketType**

the TicketType class is a subclass of the Ticket Class that will create an instance based on the type of ticket the user chooses. If a discounted ticket is chosen, the verifyDiscount() function will be called

# 3.2.12.1 Attributes of TicketType class

# 3.2.12.1.1 type

it could either be Adult, Child, Student, Military, or Senior. Will be data type: String

# 3.2.12.1.2 price

the price will be based on the ticket type. Will invoke setPrice() method that returns type: double

#### 3.2.12.2 Functions available in TicketType class

#### 3.2.12.2.1 verifyDiscount()

this function will interact with a third-party verification website such as ID.me to verify student, military, or senior status. If the verification fails, the user will be notified.

# **Arguments:**

This function has no arguments

# **Return Type:**

-boolean: the function will return true if the verification was returned as successful.

## **3.2.12.2.2 setPrice()**

this function will set the price based on the type of ticket the user chooses. If it's either adult or child ticket type, no verification is needed and the price will be set accordingly. If the user requests a discounted ticket type, the verifyDiscount() method will be invoked and the discounted price will be set based on the discounted ticket type. if the method returns false, the adult ticket price will be set by default.

# **Arguments:**

This function has no arguments

#### **Return Type:**

-void: this function returns nothing.

# **3.2.13 Payment**

The Payment Class will create an instance of the class whenever a reservation is created or a refund is requested in terms of canceling the reservation. The functions within this class will interact with a external payment gateway connected to the banking systems that will process payments and refunds. Security is a top priority to protect the information of the user

#### 3.2.13.1 Attributes of Payment class

#### **3.2.13.1.1** paymentID

each transaction will have a unique ID that will be logged and tracked using the data type: int. It will utilize a random number generator to create the unique ID

#### 3.2.13.1.2 paymentMethod

payment method type will be initialized based on what type of card they use for the transaction. The name of the card will be of data type: String

#### 3.2.13.1.3 transactionTime

date and time of transaction will be recorded with type: DateTime

#### **3.2.13.1.4** cardNumber

cardNumber will be sent to payment gateway with type: String

#### 3.2.13.1.5 cardholder\_name

will be sent to payment gateway with type: String

#### **3.2.13.1.6** security code

will be sent to payment gateway with type: int

## 3.2.13.2 Functions available in Payment class

# **3.2.13.2.1 wipeCardInfo()**

this function will set all of the attributes to null after the transaction is successfully completed, the attributes in the class will be set to null. The information will be recorded in the database beforehand. The Payment object can then be deleted from memory to protect user information.

## **Arguments:**

-the function contains no arguments

#### **Return Type:**

-void: the function will return nothing.

### **3.2.13.2.2 processPayment()**

this function is responsible for handling the payment transaction when a customer makes a purchase in a theater ticketing system. It will interact with the payment gateway to communicate with the banks to process the transaction

# **Arguments:**

- -customer (Customer): the customer making the payment
- -totalPrice (double): the total amount to be charged for the ticket booking including taxes
- -paymentMethod (String): payment method used for the transaction
- -paymentDetails (String, String, int): the cardholder name, card number, and security code

#### **Return Type:**

-boolean: returns true if the payment was successful, false otherwise

# 3.2.13.2.3 refund()

this function is responsible for processing refunds for a payment transaction. It will interact with the payment gateway that will communicate with the banks to process the refund.

# **Arguments:**

- -Transaction ID (String): the unique identifier of the payment transaction
- -refundAmount (double): the amount to be refunded to the customer

# **Return Type:**

-boolean: the function will return true if refund is successful, false otherwise

# **3.2.14 Receipt**

# 3.2.14.1 Attributes of Receipt class

the Receipt class will inherit all of the attributes of the Payment and will also have access to the variables in the Ticket class containing reservation ID

# 3.2.14.2 Functions available in Receipt class

#### 3.2.14.2.1 generateReceipt()

this function will generate either a printed or emailed receipt based off of the user input. If asking for an email receipt, it will prompt the user to enter their email address to be sent the receipt with all of the information. Otherwise, it will trigger the print function on their desktop to print a physical receipt

# **Arguments:**

- -paymentSuccessful(boolean): if true, will send a receipt, if false, will return a message saying payment was not processed correctly
- -deliveryType (String): this function will take in an argument of type String that the user will input as being either an email or printed receipt.

#### **Return Type:**

-void: the function will return nothing.

# 3.3 Software Architecture

# 3.3.1 Software Architecture Diagram

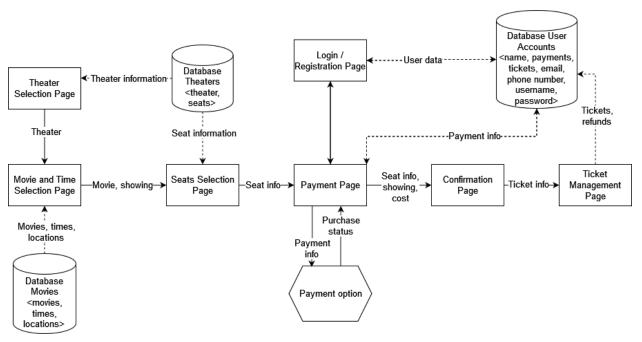


Figure 2: Software Architecture Diagram of the Theater Ticketing System including components and connectors

# 3.3.2 Software Architecture Description

The next section will provide a complete explanation of the entire SWA diagram, including an overview of components and their connectors.

# 3.3.2.1 Theater Selection Page

**Description**: On this page the user is selecting one of the theaters of the client from the San Diego Area.

**Functionality**: Selecting the theater enables the user to go further in the process of ticket buying. Once a theater is selected it will be the default theater shown in the drop-down selection.

**Connectors**: Theater Selection Page gets its theater information from the theater database which stores information about the theaters that can be selected. Once the theater is selected the user is enabled to select the movies and showtimes that are available in the theater.

## 3.3.2.2 Movie and Time Selection Page

**Description**: On this page the movie and showtime for the selected movie will be selected.

**Functionality**: Every movie currently available in the selected theater has different showtimes available. Showtimes that are booked out can't be selected and should be using a disabled secondary button.

**Connectors**: The movie database provides the Movie and Time Selection Page with data about available movies, the showtimes and the location where the movie is shown inside the theater that was selected on the Theater Selection Page.

# 3.3.2.3 Seats Selection Page

**Description**: Seats for the selected movie and showtime can be selected on this page

**Functionality**: This page allows the user to select how many seats he wants to purchase. The default value is 1 and can be increased and decreased with a plus and minus sign UI element or through clicking and writing an integer between 1 and 20 into the input field where the default value of 1 is shown. The maximum number of seats available per purchase are limited to 20.

**Connectors**: With the input from the Movie and Time Selection Page with the movie and showtime and the seat information from the Theater Database the seat selection is enabled. The seat information from this page will be directed to the next page for the payment.

# 3.3.2.4 Payment Page

**Description**: On the Payment Page the user can see his previous selections in the process and pay for the selected tickets.

**Functionality**: The Payment Page enables the user to see the selected theater, movie, location, showtime and seat information. Different payment options are available to choose from. Once a payment is confirmed the payment provider checks the validity of the payment and returns the payment status. If there is any complication the payment will be declined with an error message, if everything is correct the payment will be confirmed. Besides the option of proceeding as a guest the user can register or log in with an account.

**Connectors**: Through the seat selection all information about the purchase is displayed. The Registration / Login Page is connected to the Payment page as well. The payment option gets the payment request with the payment information and returns the payment status as confirmed or declined. The purchase information with the costs and tickets will be forwarded to the Confirmation Page. If an user account exists the payment info will be stored in the User Account Database.

#### 3.3.2.5 Login / Registration Page

**Description**: The Login / Registration Page enables the user to sign up or log in with an user account.

**Functionality**: When proceeding with the payment process there is the option to sign up or log in with a user account. The user data is stored in the User Account Database. Name, username and password are required fields but phone number is an optional input. The username will be an email address and the password has to be at least eight characters including minimum one number, uppercase letter, lowercase letter and a special character.

**Connectors**: There is a connection between the Login / Registration page and the Payment Page. When logging in with the user account the previously used payment option will be set as a default. All the user data from the registration will be stored in the User Account Database.

# 3.3.2.6 Confirmation Page

**Description**: After a successful purchase of tickets the Confirmation Page will be displayed summing up the purchase information.

**Functionality**: The Confirmation Page will display the theater, movie, showtime, seats and the location inside the selected theater where the movie is shown. Also the seat information and the costs will be shown on this page.

**Connectors**: The information of the purchase and tickets comes from the connection to the Payment Page. The ticket information will be passed on to the Ticket Management Page.

# 3.3.2.7 Ticket Management Page

**Description**: The Ticket Management Page will display the purchased tickets with the information about the purchased tickets. Also the option of refunding will be found on this page.

**Functionality**: This page enables the user to view his tickets with the ticket information (seats, movie, showtime and location) and can be used to show the purchased tickets at the theater for entry. Also there is a button that enables the user to refund the tickets.

**Connectors**: The purchase information is retrieved from the Confirmation Page. The connection to the User Account Database facilitates the process of refunding as the ticket information from the user is stored in the database as well.

# 3.3.2.8 Payment Option

**Description**: After the request for the payment the payment provider checks the payments validity and returns a purchase status.

**Functionality**: The request for the payment from the Payment Page will be checked from the payment provider and will return a confirmation or decline the payment. The status of confirmation or declinement will be returned to the Payment Page with a corresponding message on the UI of the Payment Page.

**Connectors**: The Payment Option is only connected to the Payment Page getting the payment request and returning the payment status from the payment provider.

#### 3.3.2.9 Database Theaters

**Description**: The Database for theaters stores information about theaters of the client together with seat information.

**Functionality**: The theater database provides data about the theaters for the Theater Selection Page and seat information to the Seats Selection Page.

**Connectors**: see 3.3.2.9 Functionality.

#### 3.3.2.10 Database Movies

**Description**: The movie database stored data about the movies, showtimes and locations (halls) where the movie is shown.

**Functionality**: The movie database provides the Movie and Time Selection Page with data about availability of movies in the selected theater together with showtimes for each movie and locations inside the theater where the movie is shown.

**Connectors**: see 3.3.2.10 Functionality.

#### 3.3.2.11 Database User Accounts

**Description**: The user accounts database stores information about user accounts and related activities such as payments.

**Functionality**: From the registration page the user data like name, username, password, email address and phone number is retrieved. Payment information comes from the connection to the Payment Page. Previously stored payment information can also be retrieved from the database to the payment page when the user logs into the user account. Refunds that are requested in the Ticket Management Page are checked with the information in the user account database.

**Connectors**: see 3.3.2.11 Functionality.

# 4. Development Plan and Timeline and Partitioning of Tasks

#### 4.1 Timeline

# 4.1.1 Phase 1: Project Initiation (Project Manager, Business Analyst)

Timeline: Weeks 1-2

# 1. Project Start (Week 1)

- Define scope and objectives
- Hire project manager and create development team
- Obtain deliverables
- Resource Allocation

# 2. Requirements Gathering (1-2)

- Document requirements for theater ticketing system
- Create features and functionalities

# 4.1.2 Phase 2: Design Planning (System Architect, Database Designer, UI/UX Designer)

Timeline: Weeks 3-6

# 1. Software Architecture (3-4)

- Create software architecture design
- Create software architecture diagram

# 2. Database (4-6)

- Define UML Class Design
- Create UML Class Diagram
- Plan for securing data

# 3. User Interface (4-6)

- Plan for visual interface and layout
- Gather feedback on UI

# **4.1.3** Phase 3: Development (Frontend/Backend Developers, Quality Assurance/Test Engineers, DevOps Engineers)

Timeline: Weeks 7-23

# 1.Frontend Development (Weeks 7-13)

- Develop the user interface based on UI designs.
- Implement user authentication and authorization.

# 2. Backend Development (Weeks 13-19)

• Build the server-side application.

- Implement payment processing.
- 3. Database Implementation (Weeks 19-21)
- Create the database using the planned design
- Implement data storage and retrieval functionalities.
- 4. Testing and Quality Assurance (Weeks 21-23)
- Conduct testing
- Fix bugs experienced during testing
- Make sure the system is secure

# 4.1.4 Phase 4: Deployment (Training Specialist, Support & Maintenance Team, Marketing & Promotion team)

Timeline: Weeks 24-27

- 1. Deployment and System Integration (Weeks 24-26)
- Deploy the ticketing system to production servers
- Integrate with third-party services (IMDB, Rotten Tomatoes)
- Initiate performance testing
- 2. Training and Documentation (Week 26)
- Train the theater staff on how to use the system
- Create user manuals and documentation
- 3. Soft Launch and Feedback Gathering (Week 27)
- Perform a soft launch with audience being stockholders/family
- Gather feedback

# 4.1.5 Phase 5: Launch (Support/Maintenance Team, Marketing & Promotion Team, Product Manager)

Timeline: Weeks 27-30

- 1. Full-Scale Launch (Week 27)
- Launch the Theater Ticketing System to the general public
- Promote the new ticketing system through advertising
- 2. Monitoring and Maintenance (Weeks 28-30)
- Regularly monitor the system performance and keep obtaining feedback
- Address any issues and release regular system updates (security, bug fixes, etc.)
- Plan for weekly to bi-weekly maintenance and support for the system

# Homework 3 start

# 5. Verification Test Plan

# 5.1 Introduction

# **5.1.1 Purpose**

The purpose of this test plan is to outline the overall strategies, objectives and approaches to verifying the functionality, performance, security, and usability, and compatibility of the theater ticketing system.

# **5.1.2** Scope

This test plan incorporates comprehensive testing of all aspects of the theater ticketing system, traversing all aspects of software verification.

# 5.1.3 Objectives

The main objectives of the verification test plan are to ensure the theater ticketing system:

- Meets the standards of both functional and non-functional requirements
- Performs under a varying number of loads to ensure reliability
- Maintains the highest level of security and confidentiality of data
- Provides the user an exceptional experience accessing the theater ticketing system
- Is able to function across all varying devices and platforms

# **5.2** Test Objectives

# 5.2.1 Unit Testing

- Focus on verifying individual code components that include: methods and classes
- Test cases will ensure that each unit of code performs to standard

# **5.2.2 Functional Testing**

- Verify that all functional requirements: user registration, show selection, booking a reservation, and payment processing are fully satisfied
- Validate user and system interactions along with the handling of error exceptions

# **5.2.3 Performance Testing**

• Evaluate the performance and response times of the theater ticketing system under normal and stress conditions

# **5.2.4 Security Testing**

- Identify and mitigate security vulnerabilities, such as authentication issues, data encryption, and protection against common security threats.
- Ensure that sensitive user data, including payment information, is handled securely.

# **5.2.5** Compatibility Testing

- Ensure that the system functions correctly on different web browsers (e.g., Chrome, Firefox, Safari, Edge)
- Verify that the system adapts to different screen sizes

# **5.3** Testing Approach

# 5.3.1 Unit Testing

# 5.3.1.1 Seat Reservation Functionality

**Objective:** We want to verify that the "reserveSeat" method accurately reserves the seats while updating the availability of the seat in real-time

# **Features Covered:**

- Seat Reservation Functionality
- Functionality of seat availability

- 1. Test Vector 1: Reserve a single seat and confirm it is marked as occupied
  - Input: Valid Seat Identifier
  - Expected Output: The seat is successfully reserved

# **5.3.1.2** Calculating the Total Price Functionality

**Objective:** Ensure that the calculatePrice() function correctly calculates the ticket price based on ticket type and movie.

# **Features Covered:**

- Ticket Price Calculation
- Handling of the different ticket types

# **Test Description:**

- **1. Test Vector 1:** Calculate the ticket price for a single adult seat
  - Input: Movie Type, Ticket Type: standard adult ticket
  - Expected Output: The function returns the correct ticket price for a single adult ticket
- **2. Test Vector 2:** Calculate the ticket price for a single discounted seat for a student and veteran
  - Input: Movie Type, Ticket Type: student discounted ticket / veteran discounted ticket
  - Expected Output: The function returns the correct ticket price for a discounted student / veteran ticket

# **5.3.1.3** Find Reservation Functionality

**Objective:** Ensure that the getReservation() function in the Customer Class correctly searches the database for a user's successfully booked reservation

#### **Features Covered:**

• Efficient Search Algorithm for the Database

# **Test Description:**

1. Test Vector 1: Search for a valid reservation that was previously booked

- Input: Ticket ID: valid and unique identifier, Customer last name
- Expected Output: The function correctly returns the information for the valid booking including: date purchased, total price, movie and showtime information, name of purchaser
- **2. Test Vector 2:** Search for an invalid reservation
  - Input: Ticket ID: invalid identifier, Customer last name
  - Expected Output: The function returns an error message to the user saying the reservation was not found. It will prompt the user to input a different reservation ID

# **5.3.2 Functional Testing**

# 5.3.2.1 Booking a Reservation

**Objective:** We will test the end-to-end booking process from: movie selection, showtime selection, seat reservation, and payment processing.

# **Features Covered:**

- User and system interactions including workflows
- End-to-end booking process

- **1. Test Vector 1:** Select a movie, choose showtime, reserve seats, and complete payment
  - Input: Valid movie, showtime, and seat selection and successful payment
  - Expected Output: The reservation is successfully booked, marked by a confirmation email being sent by the system
- 2. Test Vector 2: Attempt to reserve seats that are already occupied
  - Input: User attempts to book a seat that is already reserved
  - Expected Output: The system prevents a double-booking scenario and outputs an error message to the user.
- 2. Test Vector 3: User cancels a booking and receives a refund

- Input: User requests to cancel a reservation
- Expected Output: The system successfully cancels the reservation, seats are released and marked as available, and a refund is issued

# **5.3.2.2** Filtering Showtimes

**Objective:** Ensure that the system correctly filters and displays movies along with their showtimes based on the preferences set by the user

# **Features Covered:**

- Efficient Search Algorithm for the movie database
- Filtering of movies

# **Test Description:**

- **1. Test Vector 1:** Filter movies by movie rating (G, PG, PG-13 etc.)
  - Input: Movie rating selection by the user
  - Expected Output: Only movies with the specified rating are displayed along with the showtimes for that specific date.
- 2. Test Vector 2: Filter movies by both movie rating AND genre
  - Input: Movie rating selection and genre selection by the user
  - Expected Output: Only movies with the specified rating and genre are displayed

# **5.3.3 Performance Testing**

# **5.3.3.1** Multiple Users Try Booking at the Same Time

**Objective:** We will simulate multiple users attempting to book the same seat simultaneously

# **Features Covered:**

- Simultaneous Booking
- Reservation Handling

- **1. Test Vector 1:** Have multiple users attempt to book the same seat for the same movie and showtime
  - Input: Multiple users online trying to book the same seat
  - Expected Output: The system is able to handle concurrent bookings and won't allow for double booking. It should have users choose a different seat if the seat is already in the process of being purchased
- **2. Test Vector 2:** Stress Test the system with multiple concurrent booking attempts
  - Input: Very high load of concurrent booking requests
  - Expected Output: The system is still responsive and avoids crashing or system degradation

# **5.3.3.2** System Integration with Payment Gateway

**Objective:** We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions

# **Features Covered:**

- External payment gateway integration
- Payment processing and confirmation

- **1. Test Vector 1:** Perform real sample payment transactions with real payment methods customers will actually use once system is launched
  - Input: Book a movie ticket with credit card transaction
  - Expected Output: The payment is processed successfully, and the sample user receives confirmation receipt of payment
  - 2. Test Vector 2: Test for payment failures and error handling
    - Input: Book a movie ticket with debit card with zero balance
    - Expected Output: The system handles the payment failure successfully and provides the user with the appropriate error message

# **5.4 Test Cases**

see attached file "Theater Ticketing System Test Cases"

	Theater Ticketing System Test Cases									
TestCaseId	Component	Priority	Description/Test Summary	Pre-requisites	Test Steps	Expected Result	Actual Result	Status	Test Executed By	
							User was able to select movie, select showtime,			
						The seat is	reserve seat.			
					1. Select movie	successfully	Attempt to			
					2. Choose showtime	reserved and	reserve an already			
				User has browser	3. Reserve open seat	reservation of	reserved seat			
			Reserving a single seat	open AND already	4. Try to reserve an occupied seat	an occupied	wasn't possible,			
	Seat Reservation		and confirm it is marked	on client's	5. Validation that reservation isn't	seat isn't	error message to			
UT-SRF-1	Functionality	P0	as occupied	webpage	possible	possible.	user was shown.	Pass	Emre Yikilmaz	
						Payment Page				
					1. Select movie	correctly	The price on the			
				User has browser	2. Choose showtime	calculates the	Payment Page was			
	Calculating Total			open and access	3. Chose seat	ticket price for a	displayed correctly			
	Price for		Calculate the ticket price	to the ticketing	4. Chose single adult ticket	single adult	for the selected			
UT-CTPF-1	Functionality	P0	for a single adult seat	webpage	5. Proceed to Payment Page	ticket.	single adult ticket.	Pass	Emre Yikilmaz	
				User has browser		The Payment	The price on the			
				open and access	1. Select movie	Page shows the	Payment Page was			
				to the ticketing	2. Choose showtime	correct ticket	displayed correctly			
				webpage with an	3. Chose seat	price for a	for the selected			
	Handling of		Calculate the discounted		4. Chose single discounted student /	discounted	single discounted			
	different ticket		ticket price for a student /	a student /	veteran ticket	student /	student / veteran			
UT-CTPF-2	types	P0	veteran	veteran status	5. Proceed to Payment Page	veteran ticket.	ticket.	Pass	Emre Yikilmaz	
						The function	Maria Patricia			
						correctly returns	Valid ticket was			
						the information	found through the			
						for the valid	ticket search			
				Employee or	1 Admin or ampleyed aggrees to	booking	function in the			
				admin access to	1. Admin or employee access to	including: date	admin / employee			
				ticketing system with ticket ID:	ticketing system 2. Opening search function for tickets	purchased, total price, movie	access. Information about			
				valid and unique	3. Enter valid ticket ID	and showtime	the booking and			
	Find Reservation		Search for a valid	identifier and	4. Select the ticket	information,	the customer was			
	Functionality - Find		reservation that was	customer last	5. Check the customer last name and	name of	found in the			
UT-FRF-1		P1	previously booked	name	ticket information	purchaser.	system.	Pass	Emre Yikilmaz	
0111111	Valia reservation		previously booked	name	derect information	par chaser.	Invalid ticket ID	1 433	EIIIC TIKIIIIdz	
							didn't result in			
							finding a valid			
						The function	ticket in the			
						returns an error	system. Error			
				Employee or		message to the	message was			
				admin access to		user saying the	shown to			
				ticketing system		reservation was	employee / admin			
				with ticket ID:	1. Admin or employee access to	not found. It	saying the			
				invalid and unique	ticketing system	will prompt the	reservation ID			
1				identifier and	2. Opening search function for tickets	user to input a	was not found,			
	Search for invalid		Search for an invalid	customer last	3. Enter invalid ticket ID	different	asking for an valid			
UT-FRF-2	Reservation	P1	reservation	name	4. Ticket can't be found	reservation ID.	ID.	Pass	Emre Yikilmaz	

									+
							User was able to		
							select movie,		
						The reservation	select showtime,		
						is successfully	reserve seats, and		
					1. Select movie	booked, marked	complete		
				User has browser	2. Choose showtime	by a	payment.		
			Verify that user can	open AND already	3. Reserve open seat	confirmation	Confirmation		
	Booking		complete all steps	on client's	4. Complete payment	email being sent	email was		
FT-BR-1	Reservation	P0	required to book a movie	webpage	5. Receive confirmation email	by the system	received.	Pass	Jose Urrutia
						The system will	The system		
						prevent double	prevented double		
					1. Select movie	booking	booking of a seat		
				User has browser	2. Choose showtime	scenario and	that was already		
				open AND already	3. Attempt to reserve occupied seat	output error	occupied. Error		
	Booking		Attempt to reserve seats	on client's	4. System prevents double booking	message to the	message was		
FT-BR-2	Reservation	P0	that are already occupied	webpage	5. Output error message to user	user.	output to user.	Pass	Jose Urrutia
		† -	and and an eddy occupied		1. Click on "request cancellation"	1	The system		
					2. Confirm cancellation request	The system	successfully		
					3. System will check if cancellation	successfully	cancelled the		
				User has browser	window is still open (2 hours prior to	cancels the	reservation, seats		
				open AND already	movie start)	reservation,	were released and		
				on client's	4. System will cancel reservation	seats are	marked as		
					5. Seats will be released for new	released and	available, refund		
				found their	bookina	marked as	was issued, and		
	Booking			reservation on	6. Refund will be issued	available, and a	confirmation email		
FT-BR-3	Reservation	P1	Cancel a booking	client's webpage	7. Confirmation email will be sent	refund is issued	was received	Pass	Jose Urrutia
11-010-5	Reservation	P1	Cancer a booking	Client's webpage	7. Commination email will be sent	Terunu is issueu	User selected	Pass	Jose Offulia
						Only movies	desired rating and		
						with the	_		
					1 Hannell state on WElter bull antique		the system		
					1. User will click on "Filter by" options	specified rating	displayed only		
					and select "rating"	are displayed	movies with the		
					2. User will click on desired rating (G,	along with the	desired rating		
				open AND already	PG, PG-13, R)	showtimes for	along with the		
ET EC 4	Filtering		l	on client's	3. System will display movies only in	that specific	showtimes for		
FT-FS-1	Showtimes	P1	Filtering by rating	webpage	specified rating	date.	that specific date	Pass	Jose Urrutia
						The system is			
						able to handle			
						concurrent			
						bookings and	System allowed		
					1. Users will try to reserve the same	won't allow for	only one user to		
					seat concurrently	double booking.	reserve seat.		
					2. System will only allow one person	It should have	Other users		
					to reserve the seat based on	users choose a	attempting to		
					time-stamp of request	different seat if	concurrently book		
				User has browser	3. Users who were not allowed to	the seat is	the same seat		
	Multiple users try			open AND already	reserve the already reserved seat will	already in the	were prompted to		
	booking at the			on client's	receive error message and be	process of being	choose a new		
PT-MUTB-1	same time	P0	Simultanious Booking	webpage	prompted to select new seat	purchased	seat.	Pass	Jose Urrutia

The system should be able to handle a heavy stress test workload while still maintaining a multiple users try booking at the PT-MUTB-2 same time  PT-MUTB-2 System Integration with Payment  System Integration with Payment with	The system should be able to handle a heavy stress sets the system's ability to handle a heavy looking at the same time PO  System Integration with Pyment Gateway PO  We will validate the system's integration with the external payment gateway system by conducting real sample youngent failures and error moving the fact of the payment failures and error moving the fact of the payment of with Payment with Pa						•	1				
Multiple users try booking at the PT-MUTB-2 same time P0 We will validate the system's integration with the external payment gateway system by Conducting real sample with Payment with Payment should be able to handle a heavy stress should be able to bandle a heavy stress stext workload will be to be to browser on the theater ticketing system should be able to bandle a heavy stress text workload will estill maintaining a duick response times, and avoids crashing, and avoids system website website website will eld confirm payment and verify their reservation website size of the confirmation of the system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system in a web browser on the theater ticketing system on the theater ticketing system on the theater ticketing system on the degradation of the gardation of the gardation of the gardation of the gateway is functioning properly and the processed successfully which is marked by a confirmation email being sent to the browser on the theater ticketing system on the part of the theater ticketing system on the theater ticketing system on the thea	The system concurrent serservation booking from a large volume of users while maintaining a maintaining a woilds crashing and avoids crashing and									The system is able		
Multiple users try booking at the PT-MUTB-2 Same time P0 We will validate the system's integration with Payment System Integration with Payment System by with Payment with Pa	Multiple users try booking at the same time PO System Integration with Payment Gateway PO System Integration with Payment System Integration System In									to handle every		
Multiple users try booking at the PT-MUTB-2 same time P0 System Integration with Payment System Integration with Payment System In a web browser gateway system by Conducting real sample conducting real samp	Users have their web browser open on the theater ticketing system in a web browser open of the same time PT-MUTB-2  Multiple users try booking at the same time P0  We will validate the system's ability to handle a heavy load of concurrent user bookings  We will validate the system's ability to handle a heavy load of concurrent user bookings  We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  System Integration with Pyment Gateway P0  PT-SIPG-1  System Integration with Pyment Gateway P0  System Integration with payment and with Payment and website  Users have their web browser open on with Payment and verify their reservation  1. Users will open the theater ticketing mayment and verify their reservation  1. Users will open the theater ticketing system in a web browser  2. User will browser shows, select showtime and select seats (all done at the same time).  1. Users will open the theater ticketing system in a web browser.  2. User will browser shows, select showtime and select seats (all done at the same time).  3. Users will bornser shows, select showtine, and select seats (all done at the same time).  4. User will browser shows, select showtine, and select seats (all done at the same time).  5. User will browser shows, select showtine, and select seats (all done at the same time).  5. User will browser shows, select showtine and select seats (all done at the same time).  5. User will browser shows, select showtine and select seats (all done at the same time).  6. User will prowser shows, select showtine and select seats (all done at the same time).  7. User will browser shows, select showtine and select seats (all done at the same time).  8. User will browser shows, select showtine and select seats (all done at the same time).  8. User will browser shows, select showtine and select seats (all done at the same time).  9. User will browser shows, select showtine and select seats (all done at the same time).  1. User swill pome the t								The system	concurrent		
Auditiple users try booking at the same time   PO	Multiple users try booking at the PT-MUTB-2 same time PT-SIPG-1 Gateway PT-SIPG-1  We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  System Integration with the user boking a movie ticket with a debit card with zero bokang with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with Payment  System Integration with the user boking a move with the user boking a move ticket with a debit card with zero balance with the determination and system in a web browser open on the payment and verify their reservation with the user boking a move ticket with a debit card with zero balance with zero balance with the user boking a move ticket with a debit card with zero balance with zero day appropriate with select and successfully appropriate with the user obalance with the user obalance with zero balance with zero appropriate appropriate with select and information of a user seasons takes the sates (all done at the same time). User will be on the page to pay for their movie, should handle the payment to gateway is functioning properly and the payment on the paymen								should be able	reservation		
Users have their web browser open on the theater ticketing system in a web browser shows, select sability to handle a heavy load of concurrent user booking at the PT-MUTB-2 same time P0 System Integration with Payment Integration Integration with Payment Integration with Payment Integration Integr	Users have their web browser open on the theater ticketing system on the theater ticketing sys								to handle a	booking from a		
Users have their web browser open on the theater ticketing system in a web browser shows, select same time  PT-MUTB-2  Multiple users try booking at the same time  PO  We will validate the system's integration with Payment  System Integration with Payment  PStress Test the system's ability to handle a heavy load of concurrent user bookings  Users have their web browser open on the theater ticketing system website  Users will open the theater ticketing system in a web browser open on the theater ticketing and avoids degradation of the system.  3. They will all confirm payment and verify their reservation  1. Users will open the theater ticketing and avoids system of the ticketing system.  3. They will open the theater ticketing system on the theater ticketing system.  4. Users will open the theater ticketing system on the theater ticketing system on the theater ticketing system.  5. Users will open the theater ticketing system on the theater ticketing system on the theater ticketing system.  5. Users will open the theater ticketing system on the theater ticketing system.  5. Users will open the theater ticketing system on the theater ticketing system in a web browser  2. User will browser shows, select show, in the still maintaining fast response time. It also avoids crashing, and avoids degradation of the system.  The payment is processed successfully, and the sample to go through successfully, and the sample to go through successfully which is marked by a confirmation or receipt of being sent to the long system.  5. Users will powser shows, select seats (all done at the same time).  5. Users will prover shows, select seats (all done at the same time).  5. Users will powser shows, select seats (all done at the same time).  5. Users will powser shows, select seats (all done at the same time).  5. Users will powser shows, select showin	1. Users will open the theater ticketing system in a web browser   2. User will browser shows, select showtime, and select seats (all done about the same time)   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will all confirm payment and verify their reservation   3. They will be and a voids crashing, and avoids system of the same time).   3. They will all confirm payment and verify their reservation   3. They will open the theater ticketing system   4. The payment and verify their reservation   5. The payment is processed system in a web browser   2. User swill all confirm payment and verify their reservation   4. The payment is processed system in a web browser   2. User swill all confirm payment and verify their reservation   5. The payment is processed system in a web browser   5. The payment is processed system in a web browser   5. The payment is processed system in a web browser   5. The payment is processed system in a web browser   5. The payment is processed   5. The payment   5. Th								heavy stress	large volume of		
Multiple users try booking at the same time PT-MUTB-2 Stress Test the system's ability to handle a heavy load of concurrent user bookings  We will validate the system's integration with Payment with Payment  System Integration with Payment  Wultiple users try booking at the same time PO Stress Test the system's integration with Payment  We will validate the system's integration with Payment  Book a movie ticket with credit  System in a web browser shows, select showtime, and select seats (all done at the same time).  3. They will all confirm payment and verify their reservation  System in a web browser shows, select showtime, and select seats (all done at the same time).  3. They will open the theater ticketing system.  The payment is processed successfully, and the sample user receives is marked by a confirmation receipt of showtime, and select seats (all done at the same time).  1. Users will open the theater ticketing system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  1. Users will open the theater ticketing system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  The payment is processed successfully, and the sample user receives is marked by a confirmation receipt of receipt of receipt of showtime, and select seats (all done at the same time).  The payment is processed successfully, and the sample user receives is confirmation receipt of receipt of receipt of showtime, and select seats (all done at the same time).  The payment is processed successfully, and the sample successfully which user receives is marked by a confirmation receipt of receipt of receipt of showtime, and select seats (all done at the same time).  The payment is processed successfully, and the sample successfully and avoids and avoids and avoids and avoids system of the same time.)  The payment is processed successfully, and the sample successfully and the sample successfully and avoids so the same time.)  The payme	Stress Test the system's ability to handle a heavy load of concurrent user booking at the same time P0 bookings at the same time P0 bookings The payment system in a web browser open on the theater ticketing system website  PT-MUTB-2								test workload	users while		
Multiple users try booking at the same time P0 bookings  PT-MUTB-2  PT-MUTB-2  Multiple users try booking at the same time P0 bookings  We will validate the system's integration with the external payment with Payment  System Integration with Payment  Wall validate the system's ability to handle a heavy load of concurrent user bookings  Users have their web browser open on the theater ticketing system web browser open on the theater ticketing system on the theater ticketing system website  Users have their web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system.  The payment is processed successfully, and the sample user receives confirmation open open open open on the theater ticketing system in a web browser open on the same time).  The payment is degradation of the degradation of the system.  The payment is degradation of the system.  The payment is degradation of the system.  The payment is open in the theater ticketing system in a web browser open on the theater ticket	Stress Test the system's ability to handle a heavy load of concurrent user booking at the same time PD bookings at the same time PD bookings bookin							1. Users will open the theater ticketing	while still	maintaining a		
Multiple users try booking at the same time P0 bookings  PT-MUTB-2  PT-MUTB-2  Multiple users try booking at the same time P0 bookings  We will validate the system's integration with the external payment with Payment  System Integration with Payment  Wall validate the system's ability to handle a heavy load of concurrent user bookings  Users have their web browser open on the theater ticketing system web browser open on the theater ticketing system on the theater ticketing system website  Users have their web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system in a web browser open on the theater ticketing system.  The payment is processed successfully, and the sample user receives confirmation open open open open on the theater ticketing system in a web browser open on the same time).  The payment is degradation of the degradation of the system.  The payment is degradation of the system.  The payment is degradation of the system.  The payment is open in the theater ticketing system in a web browser open on the theater ticket	Stress Test the system's ability to handle a heavy load of concurrent user booking at the same time PD bookings at the same time PD bookings bookin							system in a web browser	maintaining fast	auick response		
Multiple users try booking at the pr-Mutb-2 same time P0 Stress Test the system's ability to handle a heavy load of concurrent user bookings web browser open on the theater ticketing system website same time).  The payment gateway is functioning properly and the system in a web browser shows, select showtime, and select seats (all done at the same time).  The payment gateway is functioning properly and the payment was able successfully, and the sample successfully which the external payment gateway system by conducting real sample conducting real sample ticket with credit itcket with credit itck	Multiple users try booking at the Stress Test the system's ability to handle a heavy load of concurrent user bookings are time.  PT-MUTB-2  System Integration with Payment Gateway  PT-SIPG-1  System Integration  We will validate the system's Gateway  PO  System Integration  We will validate the system's on the theater ticketing system website  We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  System Integration  We will validate the system's integration with the external payment gateway system by conducting real sample to the third department of the ticket with a debtic card transaction  We will validate the system's integration with the external payment gateway system by conducting real sample to the ticket with credit card transaction  System Integration  We will validate the system's integration with the external payment gateway system by conducting real sample to the ticket with credit card transaction  Discovery their reservation  1. User will pome the theater ticketing system and select seats (all done asystem degradation of the system's in a web browser  2. User will prowser shows, select, showtine, and select seats (all done at the same time).  3. They will be on the page to pay for their tickets  The payment user receives confirmation receipt of their tickets  The payment user receives in the payment transaction is the payment transaction the theater ticketing website  User is ready to pay for their movie ticket.  2. User has their web browser open on the theater ticketing website  2. User has seleced their movie, showtine, and seats and is ready for which they depend the payment that the user boking a movie ticket with a debit card with zero balance with zero balance of the ticket with a debit card with zero balance of the ticket with a debit card with zero balance of the ticket with a debit card with zero balance of the time ticket in the same ti						Users have their					
Multiple users try booking at the same time  PT-MUTB-2  Multiple users try booking at the same time  PO  Dokings  Doking	Multiple users try booking at the same time PO bookings on the theater ticketing system website on the theater ticketing system website on the theater ticketing system website on the theater ticketing system in a web browser on the system in a web successfully, and the sample user receives confirmation or in the system of the system in a web browser on the system in a web browser on the system in a web system in					Stress Test the system's						
booking at the same time  PT-MUTB-2  bookings  ticketing system website  3. They will all confirm payment and verify their reservation  3. They will all confirm payment and verify their reservation  The payment gateway is functioning properly and the payment to go through system in a web browser  2. User will browser shows, select showtime, and select seats (all done with the external payment gateway system by conducting real sample  System Integration with Payment  bookings  3. They will all confirm payment and verify their reservation  3. They will all confirm payment and verify their reservation  3. They will all confirm payment and verify their reservation  4. Users will open the theater ticketing system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  System Integration with payment  gateway system by conducting real sample ticket with credit  3. They will all confirm payment and verify their reservation  The payment is properly and the payment was able successfully, and the sample user receives confirmation receives of the payment is properly and the payment was able successfully and the sample user receives confirmation receives on firmation received is marked by a confirmation receive of being sent to the	booking at the same time  PT-MUTB-2    Dooking at the same time   PO   Dookings   Dookin			Multiple users try		1	1		J .	1		
PT-MUTB-2 same time P0 bookings website verify their reservation degradation system. Pass Jake Stonebraker  The payment gateway is functioning The payment is properly and the payment was able to go through system's integration with the external payment System Integration with Payment gateway system by conducting real sample ticket with credit  The payment gateway is functioning The payment is processed successfully, and the sample usccessfully, and the sample usccessfully which is marked by a confirmation email toket with credit 3. They will be on the page to pay for receipt of being sent to the	PT-MUTB-2 same time P0 bookings website verify their reservation degradation system. Pass Jake Stonebr The payment gateway is functioning properly and the payment system in a web browser 2. User will browser showtime, and select seats (all done at the same time).  System Integration with Payment Gateway P0 power their tricket with credit card transaction Teceipt of payment transactions and error handling with the user boking a movie ticket with a debit card with zero balance with Payment passed in the payment date with payment payment weeks to payment transaction weeksite verify their reservation in degradation system. Pass Jake Stonebr The payment and the payment payme			' '				,				
The payment gateway is functioning The payment is properly and the payment was able to go through System Integration with the external payment With Payment  System Integration with Payment  Graph and the payment was able to go through to go through the external payment Book a movie at the same time).  System Integration with Payment  System Integration with Pay	We will validate the system's integration with the external payment gateway ystem by conducting real sample payment transactions  PT-SIPG-1  System Integration with Payment  Gateway  PO  The payment gateway is functioning properly and the system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for their card transaction  User is ready to payment transactions  User is ready to pay for their movie ticket  Test for payment failures and error handling with the user boking a movie ticket with a debit card with payment  System Integration with payment  We will validate the system's integration with the system by system in a web browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for their tickets  The system still allowed the payment failure should handle the payment failure should handle the payment to go through and a treservation was booked for the successfully and provide the user with the user boking a movie ticket reservation with the user boking a movie ticket or payment to and with zero balance with zero bal	PT-MU	TB-2	_	P0		, ,	1 ' '	l '	1 -	Pass	Jake Stonebraker
We will validate the system's integration with the external payment  System Integration with the external payment was able successfully, and the sample uccessfully which is marked by a confirmation email ticket with credit  System Integration with the external payment was able successfully which is marked by a confirmation email being sent to the	We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  Gateway  P0  Discription with Payment  Test for payment failures and error handling with the user boking a movie with Payment  System Integration with Payment  We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  1. Users will open the theater ticketing system in a web browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for their their debit card transaction  The payment is processed successfully, and the sample user receives confirmation receipt of payment user  The system all allowed the payment the theater ticketing website  The system still allowed the payment the theater ticketing website  The system still allowed the payment the theater ticketing website  2. User has their web browser open on the theater ticketing selects seats (all done at the same time).  The payment is processed successfully, and the sample user receives confirmation receipt of payment user  The payment is processed successfully, and the sample user receives confirmation receipt of payment to their tickets  The payment is processed successfully, and the sample user receives to the in the same time).  The payment is processed successfully, and the sample user receives confirmation or the theater ticketing payment at the same time).  The system still allowed the payment the theater ticketing website  The payment as the same time).  The system still allowed the payment the theater ticketing payment as able to go through and a reservation with the user looking a movie ticket reservation with their debit card with zero balance with zero balance with zero balance at the same time).  The payment is processed successfully, and the same time, and the same time).  The system still allowed the payment to go the payment to go the payment to go the payment sall the proces											
We will validate the system's integration with the external payment System Integration with Payment conducting real sample	We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  Gateway  PO  Description with Payment  System Integration  We will validate the system's integration with the user looking a movie ticket with credit cand error handling with the user boking a movie ticket with a debit card with Payment  We will validate the system in a web browser  2. User will browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for their tickets  The payment was able to go through successfully, and the sample user receives confirmation confirmation or the payment receipt of payment  The system show the payment was received to payment transaction  The payment was able to go through successfully which is marked by a confirmation or freceipt of their tickets  The system show the payment or should handle the payment to go through and a reservation was booked for the user. It should have failed and showtime, and seats and is ready for payment with the user boking a movie ticket with a debit card with zero balance  1. User is ready to pay for their movie, showtime, and seats and is ready for payment was able to go through successfully, and the payment experiment was able to go through successfully and the payment experiment was about the payment of the theater ticketing website  2. User has their web browser open on the theater ticketing website  2. User has seleced their movie, showtime, and seats and is ready for payment with the user with the user with the sent a error message to the mander or message to the payment was able to go through and the sample user receives confirmation or payment receipt of the payment or should have failed and showtime, and seats and is ready for payment was able to go through and the sample user receives confirmation or payment receipt of the payment or payment is payment was a the sample user receives confirmation or payment was at the same time).  3. They will be									' '		
We will validate the system's integration with the external payment System Integration with Payment conducting real sample	We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  System Integration with Payment Gateway  P0  System Integration We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  Book a movie ticket with credit card transaction  The payment is processed successfully, and the sample successfully, and the sample successfully which is marked by a confirmation enail receipt of payment transaction  The payment is proceptly and the payment to go through successfully which is marked by a confirmation enail being sent to the payment user  The payment is processed  Successfully which is marked by a confirmation enail receipt of payment user  The system successfully and the sample successfully and the sample successfully and the payment to go through and the sample user receives confirmation or their tickets  The payment is processed  Successfully and the payment sourcessfully and the payment transaction of the payment to go through and the sample user receives confirmation or the payment user  The payment is processed  Successfully and the payment to go through and the sample user receives confirmation or the payment user  The payment is processed  Successfully and the payment to go through and the sample user receives confirmation or the payment user  The payment and the sample successfully and the payment to go through and a reservation was beleat seate (all done at the same time).  System Integration with the user boking a movie ticket reservation with the user boking a movie ticket and with zero balance with their debit card  System in a web browser shows, select showing and select seats (all done at the same time).  1. User has their web browser open on the payment of the payment of the payment to go through and a received for payment to go through and a received for payment to go through and a received for payment to go the payment of the payment of the payme									,		
1. Users will open the theater ticketing system in a web browser 2. User will browser shows, select showtime, and select seats (all done with Payment with Payment conducting real sample to conducting real sample to conduct ticket with credit to go through successfully, and the sample successfully which is marked by a confirmation confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully and the sample user receives is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation to confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully which is marked by a confirmation receipt of to go through successfully.	We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  Test for payment failures and error handling with System Integration with Payment System Integration with Payment User is ready to pay for their System Integration with Payment User is ready to pay for their System Integration with Payment User is ready to pay for their System Integration with Payment User is ready to pay for their User is ready to pay for their the user boking a movie ticket with a debit card with payment user on the theater ticketing system in a web browser shows, select seats (all done at the same time).  3. They will be on the page to pay for their tickets  3. They will be on the page to pay for their tickets  4. User is ready to payment to go through successfully, and the sample user receives confirmation receipt of payment user  4. User is ready to pay for their movie, showtime, and seats and is ready for with the user boking a movie ticket with a debit card with zero balance with the card information of appropriate message to the successfully, and the sample user receives confirmation receipt of payment to go through successfully, and the sample user receives confirmation receipt of payment their tickets  4. User is ready to payment to go through successfully, and the sample user receives confirmation or payment to the user because the payment to go through and a reservation was booked for the successfully and provide the user with the servation with their debit card with zero balance with zero balance with the card information of appropriate message to the servation was able to go through successfully, and the sample user receives confirmation or payment transaction the theater ticketing user user. It should servation the theater ticketing website and the same time).  5. User has their web browser open on the theater ticketing website and the same time).  5. User has their web browser open on the theater ticketing website and the same time).  5. U								The navment is			
We will validate the system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  System Integration with Payment with Validate the system in a web browser 2. User will browser shows, select showtime, and select seats (all done at the same time).  Book a movie ticket with credit of the page to pay for validate the system in a web browser 2. User will browser shows, select user receives is marked by a confirmation email being sent to the	We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  Gateway  P0  We will validate the system's integration with the external payment gateway system by conducting real sample payment transactions  Book a movie ticket with credit card transaction  3. They will be on the page to pay for their trickets  Test for payment failures and error handling with the user boking a movie ticket with a debit card with Payment  System Integration with Payment  We will validate the system's integration with the external payment gateway system by conducting real sample ticket with credit card transaction  Book a movie ticket with credit card transaction  System in a web browser  2. User will browser shows, select seats (all done at the same time).  3. They will be on the page to pay for their tickets  The system still allowed the payment failure should handle the payment failure should handle the payment failure showtime, and seats and is ready for payment  System Integration with the user boking a movie ticket with a debit card with zero balance  System in a web browser  2. User will browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for their tickets  The system should handle the payment failure should and receipt of payment should have failed and with the appropriate with the appropriate message to the							1 Users will open the theater ticketing	' '			
system's integration with the external payment System Integration with Payment with Payment  system's integration with the external payment gateway system by conducting real sample  system's integration with the external payment Book a movie at the same time).  2. User will browser shows, select showtime, and select seats (all done at the same time).  3. They will be on the page to pay for receipt of being sent to the	System Integration with the external payment gateway system by conducting real sample payment transactions  PT-SIPG-1  Gateway  PO  PT-SIPG-1  Gateway  PO  System Integration with the external payment gateway system by conducting real sample payment transactions  Book a movie ticket with credit card transaction  The system still allowed the payment transaction was pooked for the user boking a movie ticket with a debit card  System Integration with Payment  System Integration with Payment  With Payment  System Integration with the external payment gateway system by conducting real sample ticket with credit card with reditic card with payment gateway system by conducting real sample ticket with credit card showtime, and select seats (all done at the sample user receives confirmation receipt of the user payment payment to the payment showtime, and select seats (all done at the sample user receives confirmation receipt of the user payment is marked by a confirmation receipt of the user payment user  The system still allowed the payment failure should handle the payment failure successfully and provide the user boking a movie ticket reservation with their debit card with zero balance with the sent a error payment showtime, and select seats (all done at the sample user receives confirmation receipt of the user payment user  The system still allowed the payment failure should handle the payment failure successfully and provide the user with the servation with their debit card with zero balance with the card information of appropriate marked by a confirmation or confirmation or payment user.  System Integration with the external payment is marked by a confirmation or payment user receives confirmation or confirmation or payment is marked by a confirmation or payment user receives confirmation or payment being the sample user receives confirmation or payment being the payment user.  System Integration with the sample user receives confirmation or payment payment user.  System Integration with the sample user receives co					We will validate the		'	1.	1' '		
System Integration with Payment	System Integration with Payment Gateway P0 payment transactions  Test for payment failures and error handling with System Integration with Payment System Integration with Payment with Pay								l "	1 2		
System Integration with Payment gateway system by conducting real sample book a movie ticket with credit gateway system by conducting real sample book a movie ticket with credit gateway system by conducting real sample book a movie ticket with credit gateway system by confirmation email being sent to the	System Integration with Payment Gateway P0 gateway system by conducting real sample payment transactions at the same time).  3. They will be on the page to pay for their tickets  The system still allowed the payment to go through and a reservation was booked for the theater ticketing website  System Integration with Payment  System Integration Payment  System Integration Payment  System Integration Payment  System Integration					1 '				· '		
with Payment   conducting real sample   ticket with credit   3. They will be on the page to pay for   receipt of   being sent to the	with Payment Gateway P0 payment transactions ticket with credit card transaction their tickets  Test for payment failures and error handling with System Integration with Payment  System Integration with Payment  With Payment  Conducting real sample payment ticket with credit card with zero balance with payment ticket with credit card with payment ticket with credit card with payment to go through and a receivable payment to go through and a reservation with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  With Payment  The system still allowed the payment to go through and a reservation was booked for the successfully and provide the user with the sent a error message to the			System Integration			Book a movie			,		
	PT-SIPG-1 Gateway P0 payment transactions card transaction their tickets payment user Pass Jake Stonebr  The system still allowed the payment to go should handle through and a reservation was payment to go the theater ticketing website movie ticket  Test for payment failures and error handling with the user boking a movie with Payment ticket with a debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  Test for payment failures and error handling with their debit card with zero balance  The system still allowed the The system should handle through and a reservation was booked for the user. It should have failed and with the appropriate appropriate			, ,		, , ,		,				
IPT-SIPG-1 I Gateway IPO Inayment transactions I card transaction I their tickets Inayment I user Pass I lake Stonebraker	The system still allowed the payment to go should handle the payment to go should handle the payment failures and error handling with System Integration with Payment ticket with a debit card with zero balance of the system should handle the payment failure the theater ticketing website and error handling with their debit card with zero balance of the system still allowed the payment to go through and a reservation was booked for the successfully and provide the user with the sent a error with the sent a error message to the error message to the error message to the error message to t	PT-SIP	G-1	,	PΩ					1 -	Pass	Jake Stonebraker
	User is ready to pay for their movie ticket Test for payment failures and error handling with System Integration with Payment  System Integration with Payment  I User is ready to pay for their movie ticket ing website movie ticket ing website showtime, and seats and is ready for their debit card with zero balance  I User has their web browser open on the theater ticketing website movie ticket ing website showtime, and seats and is ready for payment with the sent a error with their debit card with zero balance  3 User has their web browser open on the theater ticketing website showtime, and seats and is ready for payment with the user with the user with the user with the appropriate allowed the payment to go through and a reservation was booked for the successfully and have failed and with the sent a error message to the	1 1 311 (	<u> </u>	Gateway	10	payment transactions	cara transaction	then tickets	payment		1 433	Jake Stonebraker
	User is ready to pay for their movie ticket reservation with Payment with Payment with Payment with Payment movie ticket reservation with Payment with Payment movie ticket reservation with payment wit									· '		
	User is ready to pay for their movie ticket reservation with Payment with Payment with Payment with Payment movie ticket reservation with Payment with Payment with Payment movie is ready to pay for their movie ticket and error handling with the user boking a movie ticket with a debit card with zero balance is ready to pay for their movie, showtime, and seats and is ready for payment and seats and is ready for payment itcket with a debit card with zero balance is neady for payment and seats and is ready for payment and seats and is read								The system			
	User is ready to pay for their movie ticket reservation with Payment with Payment with Payment with Payment of the payment failures and error handling with with Payment with Payment with Payment of the payment failure showled for the user balance of the payment failure showling a movie ticket reservation with their debit card with zero balance of the payment failure showling a movie ticket or payment showling a movie ticket with a debit card with zero balance of the theater ticketing website successfully and provide the user with the payment failure showling a movie the theater ticketing website successfully and provide the user with the payment failure showling and seats and is ready for payment with the successfully and provide the user with the card information of appropriate or payment successfully and provide the user with the successfully and provide the user with the card information of appropriate or payment successfully and provide the user with the successfully and provide the user with the card information of appropriate or payment successfully and provide the user with the successfully and provide the user with the card information of appropriate or payment successfully and provide the user with the successfully and provide the user with the card information of appropriate or payment successfully and provide the user with the user with the user with the card information of appropriate or payment successfully and provide the user with th											
	pay for their movie ticket and error handling with System Integration with Payment with Payment with Payment movie ticket with a debit card with zero balance movie ticket with a debit card payment their debit card with zero balance the theater ticketing website successfully and showtime, and seats and is ready for payment with the successfully and provide the user with the with the sent a error message to the successfully and provide the user with the sent a error message to the successfully and provide the user with the sent a error message to the successfully and provide the user with the user with the successfully and provide the user with the user with the user with the successfully and provide the user with the						llser is ready to	1 User has their web browser open on		1 -		
	Test for payment failures and error handling with the user boking a movie ticket with a debit card with zero balance  Test for payment failures and error handling with the user boking a movie ticket with a debit card with zero balance  2. User has seleced their movie, showtime, and seats and is ready for payment with the payment with the sent a error message to the						,	•				
	and error handling with the user boking a movie with Payment and error handling with the user boking a movie ticket with a debit card with zero balance 3. User submits the card information of appropriate appropriate have failed and sent a error message to the					Test for payment failures	1'''					
	System Integration with Payment the user boking a movie ticket with a debit card with zero balance 3.User submits the card information of appropriate sent a error message to the					1 ' '		·	, ,			
	with Payment ticket with a debit card with zero balance 3. User submits the card information of appropriate message to the			System Integration				1				
				, ,		1		1' '				
	1PT-SIPG-2 I Gateway IPO I that has zero halance I next to them I the dehit card that has zero halance I lerror message I liser I lake Stonehr	PT-SIPO	G-2	,	P0	that has zero balance	next to them	the debit card that has zero balance	error message	user	Fail	Jake Stonebraker