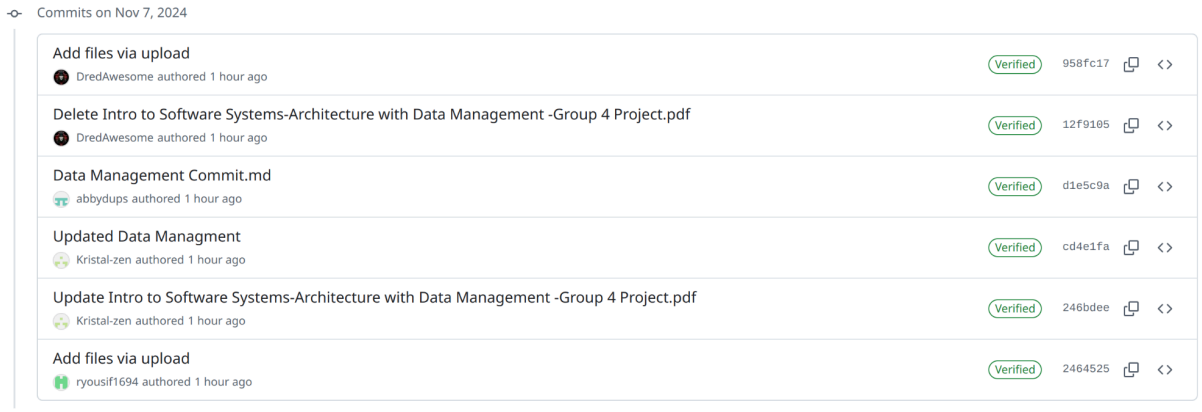
Group 4 GitHub Group URL Link for Test Plan

https://github.com/ryousif1694/Group-4?fbclid=IwY2xjawFv4gxleHRuA2FlbQIxMAABHTR_sKOjDHLmj3UOCH8mEw1hbtIvfjHe_6WAZWk6l1bG9P0fQT3Q5x7eBA_aem_vEGRvfBD3P2ZoPpxvgjwyA



Blossom Theaters

Architecture with Data Management

Version 4.0

11/5/2024

Group #4
Rita Yousif, Abigail Dupaya, Jesyl Zendejas, Judge Dred Banal

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2024

Revision History

Date	Description	Authors	Comments
<9/23/24>	<Version 1>	<Rita Yousif, Abigail Dupaya, Jesyl Zendejas, Judge Dred Banal>	<Developed Requirements >
<10/9/24>	<Version 2>	<Rita Yousif, Abigail Dupaya, Jesyl Zendejas, Judge Dred Banal>	<Added design to the document>
<10/20/24>	<Version 3>	<Rita Yousif, Abigail Dupaya, Jesyl Zendejas, Judge Dred Banal>	<Added test cases and test plan>
<11/5/24>	<Version 4>	<Rita Yousif, Abigail Dupaya, Jesyl Zendejas, Judge Dred Banal>	<Added data management, constraints and SQL diagrams>

Document Approval

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

Signature	Printed Name	Title	Date
Rita Yousif	Rita Yousif	Software Eng.	9/23/24
Abigail Dupaya	Abigail Dupaya	Software Eng.	9/23/24
Jesyl Zendejas	Jesyl Zendejas	Software Eng.	9/23/24
JD Banal	Judge Dred Banal	Software Eng.	9/23/24
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

1. Introduction

The SRS document is created to define the requirements of the movie theater ticketing system and to describe the functionalities and non-functionalities of the project. This document will provide detailed descriptions of the hardware and software along with the interface requirements that the software product of the ticketing system will have, using cases and functionalities of the ticketing system for the Blossom Theatres. This SRS document is intended to provide information regarding the software product of the ticketing system in which it will let the managers of the company know what the software product will do and what is expected from it.

1.1 Purpose

The Software Requirement Specification, SRS, provides information about the movie theater's ticketing process where customers would have the ability to attain a movie ticket online with their fingertips using our website. The reason as to why this document was made is to show the functionality of the system, information of the company and constraints that the system has on the customer's side and the employees side. The goal of this document is to give a detailed description in regards to the electronic ticketing system and provide information on how the software was used to provide this ticketing process to the customer along with what the website includes. The SRS document is built to provide information as to how the software will perform and what is expected from this software product of the movie ticketing system. Having an online ticketing system will help the customer and the employees receive information about the ticket and the customer in a more efficient manner with less time spent. This document will describe the hardware, software, interfaces and much more using cases to describe the operations of the software product.

1.2 Scope

The software product is designed for a movie ticketing theater that facilitates the online tickets that the customer obtains along with the in-person ones. The staff of the company at Blossom Theatres would be the administrators of the system to make sure that the customers of the Blossom Theatres have what they need regarding tickets, showtime, food and beverages, etc. The electronic ticketing system is called the E-Blossom Theater.

The SRS document describes the way the software is developed to provide the services to the user regarding the software system where the software product will be able to give customers the access to obtaining tickets in an electronic way rather than having to wait a long time in line to order tickets in-person. The SRS is a model that will be used in this project to define a specific standard on the functionalities of the software product. The software will have a shopping cart system in which it would give the customers the ability to obtain 20 movies in their shopping cart along with 10 different dishes and 10 beverages maximum per account. The tickets would be available to obtain two weeks prior to the premiering of the movie. This software will include movie trailers along with reviews in which this software will be able to have online reviews where these reviews will be displayed under the movie trailer. This software will also have a tab to get customer's reviews after they watch the movie regarding the movie and the service of the Blossom Theater. This software will have many interfaces including user interface, hardware

interface, software interface, and communication interface. Not only that, but the software will also have a movie search system, registering and logging in system, seating system, ticket purchasing system, administration account system, and user account system. The software system will not allow the user to enter the wrong username or password or orders more than the limited amount. The system will not allow payments or purchases from customers whose card gets to be declined.

The benefits of this software product include obtaining tickets along with food and beverages in an online platform to save time and effort rather than having to wait in line. This software system that is used for the online ticketing system of the movie theater is created in order to be a convenient way for customers to buy tickets for movies along with knowing when and where the movies would be available. This system would be done in a web browser in which it would be very easy to use for all customers who want to buy movie tickets online. The system would also allow them to choose their seats, foods, and beverages to be as satisfied as possible when watching the movie. This system is parameter driven to fulfill all the requirements to make the customer as comfortable and satisfied as possible when watching the movies in the theaters to provide them with the best possible experience.

If the SRS exists, then the software product will have a structure and a detailed description of what the software product is required to have and what the standard of the software system for the ticketing process for the movie theaters will be. If all the system requirements specifications are met, then the ticketing system would have the ability to provide a good and comfortable experience for the customers to attain movie tickets. The process of obtaining a ticket will be very easy and helpful since it contains a lot of information such as movie time, section, location, date, ticket ID number along with a QR code that will include all the necessary information to watch the movie at the theater. The software product will be in a web browser form that will include movie trailers, descriptions, pictures and all the necessary factors to know what the movie is about. The website will include upcoming movies, card information, genre tabs, customer service page, search bar, etc. The main goal of this ticketing system is to provide users with all the information they need to watch the movie and the comfortable experience regarding all the parameters and factors that are necessary to be customer friendly.

1.3 Definitions, Acronyms, and Abbreviations

FAQ	*Frequently Asked Questions*
E-Blossom Theatres	*Electronic Blossom Theatres, which is the name of the online ticketing system*
QR Code	*Quick Response Code*

ID	*Identification*

1.4 References: Examples of references that can be used

References Table

Title and Report Number	Author	Date	Publishing Organization	Website Link
IEEE Recommended Practice forSoftware Requirements Specification IEEE Std 830-1998 (Revision of IEEE Std 830-1993) SH94654	IEEE Computer Society	October 20, 1998	Software Engineering Standards Committee	Download IEEE-830-1998.pdf
List of 9 Legal Requirements for Websites and Tips to Meet Them	Masha Komnenic	January 28, 2022	Termly	https://termly.io/resources/articles/legal-requirements-for-websites/
TCP 3-Way Handshake Process	N/A	August 30, 2024	Geeks for Geeks	https://www.geeksforgeeks.org/tcp-3-way-handshake-process/
5 Main Payment Processing Compliance Regulation	N/A	January 23, 2024	SPD Technology	https://spd.tech/fintech-development/payment-processing-compliance-

				the-existing-standard-regulations-and-how-to-meet-them-best/
10 Web Application Security Best Practices You Need To Know	N/A	March 17, 2023	Medium	https://medium.com/@Imaginovation/10-web-application-security-best-practices-you-need-to-know-30a6e2fed1a2
How to Create a URL?	Avichalbharti	March 21, 2024	Geeks for Geeks	https://www.geeksforgeeks.org/how-to-create-a-url/
10 Best Domain Registrars Of 2024	Kelly Main, Rachel Williams, Peter Garcia Leets	N/A	Forbes ADVISOR	https://www.forbes.com/advisor/business/software/best-domain-registrar/

1.5 Overview

The ticket will involve a QR code that the customers could show the associates that are in the movie theater, so people can watch the movie. This process helps the customer save time because they would not be waiting in line to watch the movie. The whole goal of this document is to provide information on how the software was used to provide this ticketing process to the customer along with what the website includes. The software product is expected to have trailers, review descriptions and pictures of the movie that is being displayed or will be. The software product will have a card information system where the customer will have a one time use of card information each time they purchase in which there will be a guest purchase along with a regular customer purchase in this ticketing system. However, the regular and old customers will have their information saved about the movies, tickets and so on while the guests will not. There will be website links that are at the top of the home page in which the links will lead the customer to different pages or tabs in the browser based on their selection. The software product will include new movies that are on a monthly basis regarding the upcoming movies, an about us page, a customer service page, and a search bar. Also, there will be a genre tab that has an established algorithm that recommends movies to certain people and links where the tickets can be purchased for the movie. Not only that, but there will also be a seating limit, which is 50 tickets per theater and there will be 12 movies displayed at a time. Lastly, there will be a receipt that would be sent to email with the confirmation of the purchase of the movie ticket.

The way that the SRS is organized to give a detailed overview of what the software product will have and what it will not have. Therefore, the SRS document first has a general description of the software product then it would be related to the products that are already implemented, which is the in-person ticketing system. Also, the SRS will have the software product performance and functionalities, the characteristics of the project along with constraints, the dependencies of the software product, specific requirements that the software product would need to have to perform the required tasks, the interface requirements, non-functionalities, and inverse requirements.

2. General Description

This SRS document product will cover the ticketing system that is built in a web-browser and this is called the E-Blossom Theaters for the Blossom Theatres. This SRS document will acquire functionality and non-factionality factors about what the software will do and will not do. The software product for the movie theater electronic ticketing system will include a system for shopping cart, ticketing purchase, movie page, genre page, home page, administration and customer accounts, registering and logging in system, movie search system, seating system, and payment system. To have those systems, the software product would have to make data inputs where everything will be done in a verifiable and correct way with efficient timing. This whole process of the system operation will ease the streamline for users along with employees to checkout all the aspects of the software product where it will be easy to navigate. The SRS will be able to assist in regards to modifying factors for the project.

2.1 Product Perspective

The online ticketing system is similar to the in-person ticketing system, but way easier and more efficient to use. This is because the software product would have data input from the user regarding their preference on the movie they want and the purchasing process for the ticket, food and beverages where the ticket that they would obtain will have all the information they need. This includes ticket ID number, seating location, and QR code that would be checked by the employees at the Blossom Theater to know the movie they have purchased along with any food and beverages. The ticket would be easier for the customer to obtain in online format with simple clicks based on the customer's needs that would be saved in their account, which is located in their shopping cart. This is a more efficient way for employees to know how many people will be attending the showtime and how many intend to do so.

2.2 Product Functions

The software product will be for Blossom Theater’s customers to purchase their tickets in electronic form and software will provide the customers all the information they need to watch the movie in a more efficient way for them and for the employees too. This means that the software product will be used in a web-browser form where there will be a homepage that will include all the links to the other sites in the upper section website along with a movie search bar in which these links will be linked to other pages as well. The other pages include a ticket for the movie that the customer wants to watch in which after they select the movie they want to watch, there will be a link that will lead to another page to purchase the ticket. The page for purchasing the ticket will have a payment system embedded to it where the customer will have a one time payment. There will also be a shopping cart, so if the customer wants to watch a movie they can

add it to their shopping cart along with food and beverages and they can checkout it out from there using another link, but it will lead them to the same page for the payment. The software product will have a one time card information use regardless if the customer is a regular one or new one. The system will save information for the regular users such as shopping cart, viewed movies, reviews, purchase and much more while the guest this would not be the case. There is a limit of 20 movies that can be added to the shopping cart along with a limit of 10 food items and 10 beverages in each user's account. The ticket will have to be purchased starting from two weeks prior to the showtime of the movie. There will be 12 movies displayed at time and 50 seats maximum per room. The software product will give the customer a ticket that includes a ticket ID number, seating and room location, showtime and QR code for the information regarding what they purchased like movie ticket, food and beverages. The software will include reviews, trailers and descriptions about the movie in which this means that there will be a system that will be embedded into the software product to show the reviews to the customer and when the customer is done with watch the movie, they would receive a message to give a quick review regarding the movie and the services that would lead to reviews tab in the web browser of the website to do the reviews there. This software will also have a registering system for both users and employees accounts along with logging in. The software will also have many interfaces such as user, hardware, software, and communication interfaces.

2.3 User Characteristics

The users of the movie ticketing system will be the customers, theater floor staff, and the administrator staff of Blossom Theaters.

Customers will not have any required educational level as the system will be designed to be user-friendly, and a tutorial will be available on the website if needed. They should have basic experience using mobile and computer devices and be familiar with purchasing products online. They should have an active email and phone number, as confirmation will be sent via these channels.

The theater floor staff refers to the individuals who will be scanning the customer's tickets, checking in the customers, and helping people purchase tickets at the movie theater location. They should have at least a high school diploma or equivalent. They should have basic computer experience or be familiar with using point-of-sale systems. Little to no technical expertise is required as they will be trained to use the movie ticketing system.

Administrator staff refers to the owner, sales, accounting, marketing, and customer service departments. They should have at least a high school diploma or experience in a related field. They should have basic computer skills and be familiar with managing systems, exporting, and importing data. Moderate technical expertise is needed as they will be responsible for uploading, removing, scheduling the showtimes, and doing administrative tasks on the system. Training will also be provided.

2.4 General Constraints

The constraints of the system are the following:

a. Regulatory policies

The movie ticketing system should comply with federal and state regulation policies involving online consumer purchases. The system will comply with the California Consumer Privacy Act (CCPA), California Privacy Rights Act (CPRA), and California's Online Privacy Protection Act (CalOPPA) to protect customer information and privacy. The system will also comply with the Americans with Disabilities Act (ADA) for users who need additional website accessibility. For payment transactions, the movie ticketing system will comply with Payment Card Data Security Standards (PCI DSS).

b. Hardware Limitations

The movie ticketing system should work with mobile devices, tablets, laptops, desktop computers, and barcode or QR code scanners.

c. Interfaces to Other Applications

The movie ticketing system should be able to interact with other applications, such as payment API, payment gateway, email API, and short message service API.

d. Parallel Operations

Different users should be able to use and access the movie ticketing system at the same time. The system should also handle large traffic volume during busy times.

e. Audit functions

The system should be able to track and record user transactions on the system to ensure accountability and traceability of user actions.

f. Control functions

The system should restrict specific functions based on user roles. Administrative staff should be the only one to have access to modifying movie showtimes. Theater floor staff should be able to help customers make on site purchases and validate their tickets. Customers should only be able to buy, ask for refund, and browse movie selection. Quantity of tickets allowed to be purchased should also be limited to avoid ticket scalpers.

g. Higher order language requirements

The movie ticketing system should be developed using a higher order language that supports integration with third party APIs and cloud based services.

h. Signal Handshake protocols

A Signal Handshake protocol should be used in the system to ensure all transactions and communications between the client and server are secure. This protocol must also support reliable and secure authentication and validation of all the customer transactions.

i. Reliability Requirements

The movie ticketing system should be able to support a high volume of users and ideally be able to load its page within a maximum of 2 seconds. It should be up 99.98% in a year, the remaining 0.02% which equates to 1.752 hrs or 105.12 minutes will be spent for system maintenance and will be performed during non business hours. It should be able to handle errors entered by the users. The system should be scalable and able to add other locations in the system, in the event that the owner decides to build or buy more locations.

j. Criticality of the application

The movie ticketing system should be accessible and usable at all times during business hours. In the event of failures during transactions, the system should have backup measures in order to prevent data loss and save the customer's transaction progress.

k. Safety and Security consideration

Safety and security measures should be placed in the system to protect Blossom Theaters and customer information and data. The administrative and theater floor staff should have a password to access the system. A two factor authentication will be required for the administrative staff to ensure the security of the system and no unauthorized changes are made. There should be a way to check that the ticket barcodes and customer payment transactions are valid.

l. Budget

A maximum budget of $200,000 will be allocated in building the system. This amount will allow the system to service a large amount of users and will make it scalable in the event of expansion.

2.5 Assumptions and Dependencies

The movie ticketing system is accessible via the internet through the URL blossomtheaters.com. It is assumed that the users will have a stable internet connection and have access to widely used devices such as mobile phones, tablets, laptops, or desktop computers. Their devices should also support modern web browsers such as Safari, Google Chrome, or Firefox. Since the movie ticketing system utilizes online payment functionality, customers are assumed to have a debit or credit card.

It is assumed that the movie ticketing system will continue to be compatible with or supported by Stripe, the third-party payment gateway that the system will use. If the system is no longer

compatible with Stripe, the movie ticketing system must find another provider to continue offering online payment transactions.

The system is dependent on a cloud server database provider. Any disruptions to this will halt the system's operation, and there will be no other way to access the data.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface must be visually appealing and user-friendly. Users should be able to interact with the system on widely used devices, such as mobile phones, tablets, laptops, and desktop computers. The system should adjust the website layout automatically to fit the screen size of the user's device. The images, buttons, colors, and fonts of the system should be displayed properly on Safari, Google Chrome, and Firefox. The customer, theater floor staff and administrative staff will have different interface designs and capabilities tailored to their specific needs and roles.

The user interface should use visual images and appropriate colors based on the theme of the movie to create a more engaging user experience. There should be buttons that will take the users to their desired landing pages and actions, such as browsing available movies, purchasing tickets, choosing their seating, and filtering search options based on genre, theater location, or rating. If a user makes an error or inputs incorrect information, a pop up message should appear, informing the user of the specific error and providing instructions on how to address the issue. There will be a home page where the users will see the other options to take them to the system's different landing pages.

3.1.2 Hardware Interfaces

The movie ticketing system should support the integration of barcode or QR code scanners to validate and scan the tickets. The scanner should be able to connect with the software system via a USB cable or wirelessly via Bluetooth. After scanning the code, the software system should validate the successful receipt of the ticket and update that it has been claimed.

3.1.3 Software Interfaces

The movie ticketing system should be able to integrate with the following software products:

1. A Cloud Based Database will store customer information, transactions, sales data, seating information, and available movies.

2. Stripe will be used to handle payment gateway and processes. By using Stripe, the system will be able to support, authenticate, safely process, and receive online and in-person payments using different payment methods.

3. The movie ticketing system should support Windows, Mac IOS, and Android Operating systems to be accessible to any customer's device.

4. An email and phone notification system API should be used to notify the customers of their transactions and confirm via email and SMS.

3.1.4 Communications Interfaces

The movie ticketing system will require a stable connection to the internet. Communications between the server and web browser will use Hypertext Transfer Protocol Secure (HTTPS) to ensure a secure connection and protection of sensitive data like the user information and payment details. For the hardware devices, a USB interface will be used to communicate between the computer and the QR code scanner. This will allow the movie ticketing system to scan and validate the tickets.

3.2 Functional Requirements

URL Registration:

1. Chose a domain name: Blossom Theaters

2. Registering the domain Name/ Selecting a hosting provider: Use a domain registering platform, in this case we will use IONOS

3. Link your domain to your hosting: Configuring domain name

    a. log into the IONOS account, follow the instructions given on page

3.2.1 Registering

3.2.1.1 Introduction

- Registering will be a feature that allows users coming onto the website to create an account. This will allow access to purchasing tickets and information saving like name, email, phone number, date of birth and ticket booking history.

3.2.1.2 Inputs

- Username: unique to every user

- Password: Has to meet certain requirements

    - Has to obtain at least 8 characters, one uppercase, one lowercase, 1 number and 1 special character.

- Email address: valid email account that is verified and that has not been used before

- Phone number: stores phone number, to allow login verification

- Date of birth: will store in order to suggest age appropriate movies

3.2.1.3 Processing

1. User will be lead to a page for registering, this page obtains all boxes to be filled for information
2. User fills the information
3. The system will process and validate if all inputs do not meet criteria (in terms of being unique and quantity requirement being met)
4. If everything checks out then a message is sent to the email and phone number

3.2.1.4 Outputs

- Success Page: A new screen will pop up having a message that will say ‘Account made successfully, please verify your account through email and log in!’
- Confirmation Email: A confirmation email being received for authorizing an account.

3.2.1.5 Error Handling

- Will display errors if one requirement is not filled or in an incorrect format.
    - Also displays a message of what is filled wrong

3.2.2 Logging in

3.2.2.1 Introduction

- This allows existing users to log in and access information as well as edit information (except for certain things like date of birth).

3.2.2.2 Inputs

- Username: Unique username
- Password: the password that matches with said username

3.2.2.3 Processing

1. User will enter information into the appropriate boxes
2. System will take inputs and validate the inputs against what is stored in the database
3. If the information does not match then the system will reload the page with a failed message
4. Successful match will equate to access to the user’s account.

3.2.2.4 Outputs

- User's Page: this means that login was successful and allows user to see their information

- Failed Login: Will indicate a failed attempt of logging in (reloads page)

3.2.2.5 Error Handling

- Will lock the account after 5 tries and will send an email to the user letting them know. Sending them a link to change their password.

3.2.3 Movie Search

3.2.3.1 Introduction

- This search option will allow users to find movies. This can do it through genre, dates (recent releases or going soon), rating filter or by movie name.

3.2.3.2 Inputs

- Movie name search: this will include the name of the movie

- Genre filter: this will be a drop down that will allow the user to pick a genre

- Date filter: dropdown that will allow you to chose movies based on whether they are new or the movies are leaving soon

- Rating filter: will be a dropdown bar that allows

3.2.3.3 Processing

1. User inputs what they want to input (don't have to fill every box)

2. System will process what the user has input for each box

3. Then system will go through database in search for movies that match the criteria

4. Results will be displayed

3.2.3.4 Outputs

- Search results: will display the movies that match the criteria, matching either the movie name, date, genre or ratings

3.2.3.5 Error Handling

- If no movies match what if input into the filter then a page that states 'No movies found with these criteria are found'

3.2.4 Seating

3.2.4.1 Introduction

- This makes it so that users are allowed to select their desired seats for a certain showtime (maxing out at 20 tickets).

3.2.4.2 Inputs

- Showtime: the time user wants to watch the movie (selecting from a list of available times)
- Desired seats: will be the selected seats of the user

3.2.4.3 Processing

1. Once the user chooses a movie they want to watch then they are directed to a page of movie times for desired day.
2. Once choice is made the system will bring the user to a new page where a seating chart is displayed
3. User selects seats
4. System reserves the seats for specific user

3.2.4.4 Outputs

- A popup will appear asking if user wants to purchase snacks ahead of time
    - If yes is selected then they will be redirected to the food page.
    - If no is selected then the Ticket Purchase page will load.

3.2.4.5 Error Handling

- If the seat is not available anymore then the page will reload and allow the user to pick other seats.
- If a user tries to choose more than 20 seats then the system will tell them it’s not allowed and they must retry with the max being 20.

3.2.5 Purchase

3.2.5.1 Introduction

- This process adds up the total and allows the user to purchase the ticket and food/ drinks that they had already preselected previously.

3.2.5.2 Inputs

- Billing address: to ensure correct information corresponds with card information.
- Card information: needed in order to clear
    - Full Name
    - Date of expiration
    - Card Number
    - CVV

3.2.5.3 Processing

1. User will fill out payment information
2. System will process the payment
3. Ticket QR code will be generated, displaying the movie name, date, time, name of visitor and showtime.

3.2.5.4 Outputs

- Ticket QR code will be curated and sent to email address
- If food is purchased then a confirmation number is sent to user

3.2.5.5 Error Handling

- If payment fails then the system will inform the user and will allow them to retry
- If payment is abandoned within six minutes then the seat reservation is canceled.

3.2.6 Booking History

3.2.6.1 Introduction

- This allows user to access past movie tickets, providing them an easy way to review the movie

3.2.6.2 Inputs

- User Login must be satisfied
    - User credentials: User must be logged in already

3.2.6.3 Processing

1. User opens the tab on the home webpage that says ‘Movie Viewing History’
2. System will retrieve the history of said user
3. System displays the movies, including information like date and time viewed.
4. User can click on movie and page will open portraying information and a text box as well as star rating in order to allow the user review
5. User input will be added to the list of other reviews.
6. System updates overall movie review

3.2.6.4 Outputs

- Users booking history list: this includes information like, movie names, showtimes, date, seat numbers and rating feature

3.2.6.5 Error Handling

- If user has not purchased movie tickets before then the page will be empty

3.2.7 Food Handling

3.2.7.1 Introduction

- This feature will allow users to purchase food while in check out for movie tickets. The purchase will be made and sent to the kitchen under the confirmation number and name of the user.

3.2.7.2 Inputs

- User choice from a selection of food and drinks
- Quantity desired
- Time of arrival

3.2.7.3 Process

1. User is directed from the seat selection page to the food page
2. User selects food/ drink item
3. User redirected to page with said food item, displaying necessary information like calories
4. User selects quantity and adds to cart

5. System will process this request and add the food/ drink to the shopping cart
6. Once purchase is made there is a ticket sent out the the kitchen with this information

3.2.7.4 Outputs

- This will display the purchase page where users input their information and continue with the purchase
    - When this is done, they then receive a confirmation number on app and through email

3.2.7.5 Error Handling

- If user tries to purchase more than what is allowed then it displays an error message, informing them of a limit:
    - popcorn - 10
    - Drinks- 10
- Error message displayed saying ‘Shopping cart full’, ‘order maximum’

3.2.8 Accounting notification

3.2.8.1 Introduction

- This feature informs the accounting department of a new sale that has been made. This will make it so that records are always kept up to date and are accurate.

3.2.8.2 Inputs

- Sales details:
    - transaction number: unique to each purchase made
    - Date and time of the sale: for reference
    - Total amount paid
    - Payment method
    - Items that were purchased: making sure they are able to account for restocking needed
- Sales person email: making it easy to be sent the information

3.2.8.3 Process

1. User makes a purchase
2. A unique transaction number is generated
3. Information of transaction is sent to department, referencing this transaction number

3.2.8.4 Outputs

- Sales email that will include:
    - transaction number: unique to each purchase made
    - Date and time of the sale: for reference
    - Total amount paid
    - Payment method
    - Items that were purchased: making sure they are able to account for restocking needed

3.2.8.5 Error Handling

- If notification fails then there is an error message displayed, prompting the system to be checked, ‘ unable to notify accounting’

3.2.9 Shopping Cart

3.2.9.1 Introduction

- Shopping cart allows users to view what items they are purchasing before finalizing their transaction.

3.2.9.2 Inputs

- Ticket Selection : Movie being viewed, showtime and seat choice
- Food and drink selection: added along with desired quantities

3.2.9.3 Process

1. Once the selections are made for the movie, showtime, seats, food and drinks then users are lead to their shopping carts
2. This is where users can change/ edit their cart, adding, deleting items; or Increasing/ decreasing the quantity of items

3.2.9.4 Outputs

- Cart Display: Cart displays all the items the user has in their cart
- Checkout: There will be a checkout button that redirects them to the payment page
- Error: if the amount of items exceed limit amount

3.2.9.5 Error Handling

- If user has an empty cart and tries to proceed then they will get an error message letting them know they need to add movie tickets or food and drink to cart
- If item is no longer available then there will be an error message informing them that it is not available, allowing them to return to shopping cart to make different a choice

3.2.10 Movie Page

3.2.10.1 Introduction

- This will be the page that holds information pertaining to a certain movie. This page will include a movie trailer, pictures of scenes from the movies, some reviews and showtimes.

3.2.10.2 Inputs

- Movie ID- unique movie ID for every movie
- User Reviews- Displays reviews from customer users
- Showtime selection- this list out all the available showtimes
- Search- this is how user can get to the movie

3.2.9.3 Process

1. User will search for movie and once movie is selected, user will be redirected to movie page
2. The system will retrieve the movie data based on the movie ID and pull up the information from the database
3. The system will then display the movie title, genre, duration, rating, release date, trailer

3.2.10.4 Outputs

- Movie details: this is the movie name genre, duration, release date and trailer
- Media: this is an embedded link with the youtube video to the trailer
- User reviews: movie reviews made by other users/ customers

- Showtimes: A list of available movie times

3.2.10.5 Error Handling

- Showtime selection issue: if the movie is full or if it's too late to purchase tickets then the page will reload, notifying the user and letting them retry with a different film.

3.2.11 Guest purchase page

3.2.11.1 Introduction

This will allow users that do not currently have an account to purchase tickets.

3.2.11.2 Inputs

- Movie selection: The movie that the user wants to view
- Showtime selection: The showtime the user chose
- Number of tickets: quantity of tickets to purchase
- Payment information
    - credit/debit card details
        - Name
        - Mailing address
        - CVV
        - Card number
        - Expiration date
    - Email address
    - Phone number
    - Customer/ movie visitor's name: The name that will be printed on the ticker

3.2.11.3 Process

1. System will first prompt user to create a new account in order to certain information
    a. If customer choses yes they will be redirected to registration page
    b. If customer says no then it will continue to guest checkout
2. The customer will be lead to the shopping cart in order to finalize payment

3. Once shopping cart is confirmed they are leat to guest purchase page
4. User will input information in correct boxes
5. System will process the payment
6. Ticket QR code will be generated, displaying the movie name, date, time, name of visitor and showtime.
7. System will display success message and lead user to a link that contains their ticket

3.2.11.4 Outputs

- Total Cost: this will be the total price for everything in users cart
- Transaction confirmation: This is what will be sent to the sales team
- QR code Ticket: This will be used to gain entry into movie theater
- Email: confirmation email with the ticket is also sent out after a successful transaction

3.2.11.5 Error Handling

- Payment process: if the payment does not go through the payment page will reload and allow the user to retry payment information entry
- The system will display a message saying exactly which field was filled in incorrectly
- Missing information: This will display an error message pertaining to missing information and will reload the page, allowing the user to fill in the box

3.2.12 Logout Page

3.2.121.1 Introduction

- This will be the default page that users land on after logging out of the system. This page will let users know that they have successfully logged out of the system.

3.2.12.2 Inputs

- User action: A push of the logout button

3.2.12.3 Process

1. User will press the log out button when they desire to log out
2. System will process this request and begin the logout process

3. System will output a ‘logout successful’ message informing the user they are now logged out

4. System will redirect user to log out page (will have different link options)

5. Allows users to go back to home screen

3.2.12.4 Outputs

- Successful logout loading screen: a message displaying ‘you have now been logged out successfully!’

- Redirection links: Links that will be displayed as buttons that will take a user back to the homepage, enabling them to still browse the website

3.2.12.5 Error Handling

- Logout error: if the customer can not log out successfully then the system will say this and redirect them to the customer service phone number

- Link misdirection: If link fails to load homepage then an error message will be displayed saying so.

3.3 Use Cases

3.3.1 Use Case #1: Registration

Allows a visitor of the page to create an account and store information, making the process faster and easier for users.

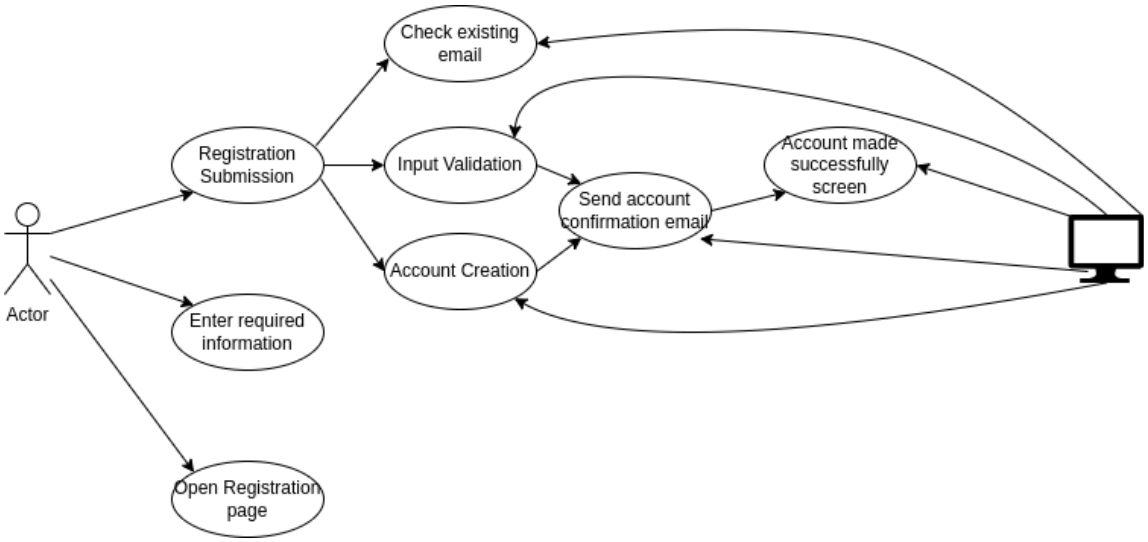
Actor: User, system

Conditions:

- **Precondition:**
    - Must have access to internet
    - Must have email
    - Must have phone number
    - Must have address
    - Must have credit/ debit card
- **Postcondition:**
    - A new user account will be created and stored in database

Steps:

1. User opens registration page
2. User enters required information such as email, phone number, password and date of birth
3. System must validate that everything is in proper format
4. System creates a new account storing said information
5. System send out confirmation email to user
6. User receives a confirmation email from the system confirming an account will be made



3.3.2 Use Case #2: Movie Search

Allows the user to search movies (pertaining to certain criteria) in order to start the booking process.

Actor: User, system

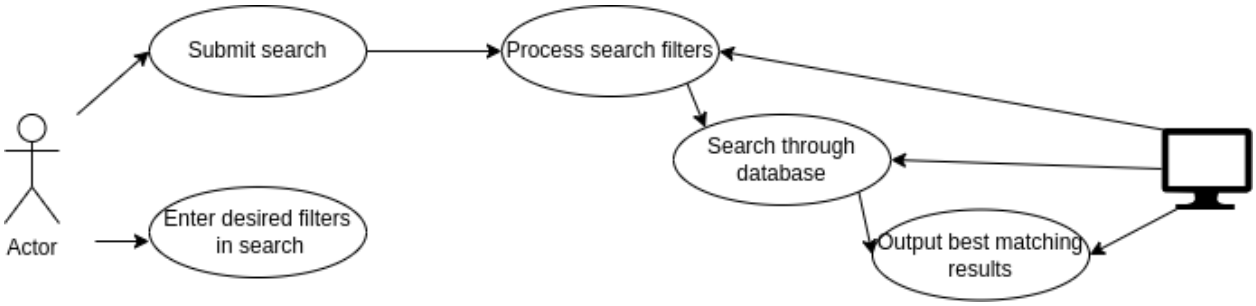
Conditions:

- **Precondition:** user must be on Blossom Movie's website in order to access search bar
- **Postcondition:** User will be able to view movies matching certain criteria

Steps:

1. User enters desired search criteria (dates, newest-leaving soon, ratings, genres or movie name)

2. System will process request
3. System will look through database picking out movies that are relevant to the search
4. System will output the results of the search



3.3.3 Use Case #3: Rate and Review

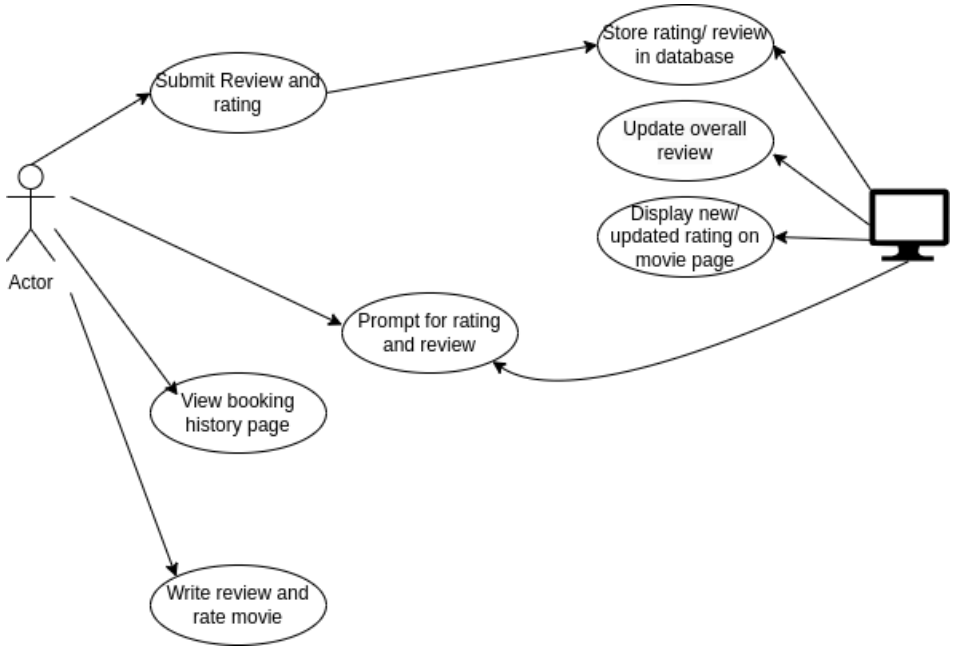
Actor: User, system

Conditions:

- **Precondition:**
    - User must already have watched the movie that they will review
    - Must be logged into an account
- **Postcondition:**
    - System will display a list of reviews on movie page that other users are able to look at

Steps:

1. User visits booking history page
2. User selects movie previously watched
3. System prompts user to rate and review the movie (on a 1-5 star system) and can write a review in the textbox
4. User submits review and rating
5. System stores this rating in database
6. System then displays this rating in movie's page
7. System updates overall rating



3.4 Classes / Objects

3.4.1 Class / Object #1: User account

3.4.1.1 (Attributes)

- ➔ User ID number- a unique # assigned to each user
- ➔ Username -name of choice to be displayed
- ➔ Password - in order to login, keeping security to users profiles
- ➔ Email - user email

3.4.1.2 (Functions)

- ➔ login() - will log an existing user in
- ➔ register() - register for new account
- ➔ purchaseHistory() - view past movie viewings
- ➔ updateInformation() - where user can update home address, password or delete account

3.4.2 Class / Object #2: Admin Account

3.4.2.1 (Attributes)

- ➔ Work ID number- makes it so has to be employee number in order to login
- ➔ Password- ensuring security
- ➔ Authentication approval- further protection (proving customer's identity)

3.4.1.2 (Functions)

- ➔ addMovie() - allows admin to add new movies
- ➔ deleteMovie()- deletes movies
- ➔ editMoviepage()- enables editing of the movie information tab (time, duration, description, trailer)

3.4.3 Class / Object #3: Movie (page)

3.4.3.1 (Attributes)

- ➔ Movie ID - unique ID per movie
- ➔ Genre - allows user to search
- ➔ Ratings- watchers opinion
- ➔ Show Times- possible times to watch movie
- ➔ Duration- how long the movie is at
- ➔ Seat Numbers- seats in the movie theater
- ➔ Movie Trailer- Movie trailer video

3.4.3.2 (Functions)

- ➔ seatsNumbers() - handles seat numbers, ensuring theater is not over capacity
- ➔ addRatings()- enables user to add review to movie
- ➔ movieDetails()- handles the movie information
- ➔ showTimes()- list available showtimes

3.4.4 Class / Object #4: Payment

3.4.4.1 (Attributes)

- ➔ Transaction ID - transaction assigned # to keep track of failed/ successful transactions
- ➔ User ID- user that the ticket is assigned to
- ➔ Total- total price for everything in shopping cart
- ➔ Payment Method- will see what the payment is made with (visa, apple pay, google pay)
- ➔ Status- fail or successful payment (did the payment go through)

3.4.4.2 (Functions)

- ➔ paymentMethod()- figures out which payment method is used and sends out request for payment
- ➔ processPayment()- sends out request to proper bank
- ➔ validatePayment()- confirms validation or card being used from the bank.
- ➔ accountingMessage()- Confirms sale was made and allows accounting to reach out to bank for further processing

3.4.5 Class / Object #5: Ticket

3.4.5.1 (Attributes)

- ➔ User ID number- a unique # assigned to each user
- ➔ Movie ID - unique ID per movie
- ➔ Show Time- movie showtime
- ➔ Seat Numbers- seats in the movie theater
- ➔ Air Date- the date
- ➔ Food/ Drink Order#- This will show the food order confirmation number

3.4.5.2 (Functions)

- ➔ storeTicket()- stores the ticket in the account
- ➔ sendTicket()- sends ticket as an email
- ➔ ticketGenerator()- generates a QR code

- ➔ mealOrder()- attaches food confirmation number to QR code

3.4.6 <Class / Object #6> Search

3.4.6.1 (Attributes)

- ➔ Genre- allows user to search through genres of movies
- ➔ Movie ID- unique movie ID to ensure right movie
- ➔ Date Filter- newest movie first
- ➔ Genre Filter- filters through genres
- ➔ Rating Filter - filters through ratings

3.4.6.2 (Functions)

- ➔ displayResults()- display search results
- ➔ displayDefault()- under search bar, displays the newest movies
- ➔ searchFilter()
    - ◆ dateFilter()
    - ◆ genreFilter()
    - ◆ ratingFilter()
    - ◆ location()
    - ◆ showtime()
- ➔ clearFilters()

3.4.7 Class / Object #7: Shopping Cart

3.4.7.1 (Attributes)

- ➔ User ID- this is the user that the ticket will be assigned to
- ➔ Cart Items -things that customer can buy
    - ◆ Food Ticket
    - ◆ Movie Tickets
- ➔ Total Price- added price of all items and the appropriate tax takeaway

3.4.7.2 (Functions)

- ➔ addItem() - add item to carts
- ➔ deleteItem()- deletes items from cart
- ➔ updateItemquantity()- updates the amount of tickets or food
- ➔ totalCalculator() - calculate totals by adding the price of individual items, then adds tax
- ➔ clearCart() - clears the cart of any items, food or movie tickets
- ➔ sendKitchenticket()- will send out ticket to the kitchen to start preparing food

3.5 Non-Functional Requirements

3.5.1 Performance

The website should load quickly and respond to user input effectively. Performance shouldn't be affected when a lot of users are logged on at once, especially during busy periods like when new movies are released. The user experience would be enhanced by aiming for a page load time of less than two seconds and quick response times of less than one second for searching and booking tickets.

- Page load time: < 2 seconds
- Max concurrent users: 800+ without slowdown
- API response time: < 1 second

3.5.2 Reliability

The website should consistently perform its functions correctly and without failure. Critical features, such as booking systems and payment gateways, should have minimal downtime. The system should be thoroughly tested to ensure it works under various conditions and can handle errors gracefully without affecting the overall experience. Additionally, a high-quality internet connection provided by a reliable ISP will ensure uninterrupted access to the server.

- Uptime: 99.98%
- Reliable Internet Provider: The system will rely on a high-speed, dependable internet connection, provided by a trusted ISP, to guarantee optimal performance.
- Error tolerance for critical operations like booking and payments.

3.5.3 Availability

The website should be accessible 24/7 with minimal downtime, ensuring that customers can book tickets, view movie schedules, and make payments at any time. To achieve this, redundancy strategies, such as backup servers and automatic failover mechanisms, should be implemented.

- Service Uptime: 99.98% or higher to minimize downtime and ensure consistent access.
- Redundancy: Use of backup servers and failover systems to prevent outages during failures.
- Scheduled Maintenance: Communicate any planned maintenance to users well in advance to minimize inconvenience.

3.5.4 Security

Implement modern security practices, such as SSL encryption, two-factor authentication, and secure payment gateways. Sensitive user data, such as payment information, should be encrypted, and the website should be regularly tested for vulnerabilities using penetration testing tools. Additionally, implementing a Web Application Firewall (WAF), robust input validation, proper user session management, and strong error handling mechanisms will further enhance security. The system should be protected from common threats like SQL injection, XSS, and DDoS attacks.

- HTTPS/TLS encryption for all data transmission
- Regular vulnerability scanning and patching
- Multi-factor authentication (MFA) for sensitive operations
- Web Application Firewall (WAF) protects the system from SQL injection, XSS, and other attacks.
- Input Validation ensures user inputs are sanitized to prevent security breaches.
- User session management uses robust session IDs, expiration dates, and monitoring to prevent hijacking.
- Error handling and logging errors are properly handled without revealing sensitive data.

3.5.5 Maintainability

The website should be built using modular and well-documented code to ensure easy maintenance. New features should be easy to implement, and updates should be performed without disrupting user operations. Using common development frameworks and best practices will make the website easier to maintain over time.

- Modular codebase with clear documentation

- Automated tests to ensure quick bug fixes
- Version control and CI/CD pipelines

3.5.6 Portability

The website should be compatible across different platforms, including mobile, tablet, and desktop. It should be easily deployable on various hosting environments, such as Amazon Web Services, Google Cloud, etc., without extensive reconfiguration.

- Responsive design for different screen sizes
- Cross-browser compatibility
- Cloud platform independence for easier scaling and migration

3.6 Inverse Requirements

- The system should not store credit card information to avoid risks related to handling sensitive financial data. Instead, payment processing will be handled by third-party services such as Stripe.
- The website should not support outdated browsers to avoid additional maintenance and security risks.
- The website should not allow anonymous bookings without user registration to ensure accountability and reduce the chance of fraudulent activities.
- The website should not offer downloadable content such as movie trailers, teasers, or videos to avoid bandwidth consumption and protect against potential copyright issues.
- The website should not include ads to maintain a clean, distraction-free user experience focused on movie booking and information.
- The website should not allow multiple user accounts using the same email to avoid account duplication and confusion.
- The system should not process incomplete or invalid booking requests, ensuring data integrity and accurate ticket reservations.
- The system should not store passwords in plain text, requiring encryption methods for password storage.
- The system should not log sensitive data in server logs for security reasons.
- The website should not expose internal error details to users, displaying only user-friendly messages.

- The website should not automatically extend user sessions indefinitely, requiring re-authentication after a set period.
- Person who enters the wrong username or password, orders more ticketing than limited amount
- People that have their card declined
- Goes over max items in shopping cart
    - Every category has their own limits
        - Tickets- 20
        - popcorn - 10
        - Drinks- 10
    - Per account

3.7 Design Constraints

3.7.1 Platform Compatibility

- The system will be designed to be compatible across both desktop and mobile platforms. It must be responsive, ensuring that all critical functionality is accessible and fully operational on devices ranging from smartphones to large desktop displays.

3.7.2 Security and Data Protection

- All sensitive information, including passwords and payment details, will be encrypted both in transit and at rest. This will include encryption standards such as SSL/TLS for secure data transmission and AES-256 for data storage.
- Role-based access control (RBAC) will be implemented to ensure only users with the correct roles (admin, regular user) can access or manipulate certain functionalities like user management, movie management, and financial transactions.
- User sessions will be securely managed and will expire after 15 minutes of inactivity to prevent unauthorized access. Admins will be required to use two-factor authentication (2FA) for access.

3.7.3 Payment Integration

- The system will be integrated with third-party payment gateways, such as Stripe, to handle user payments. The system will not handle or store sensitive payment information directly; it will delegate this task to secure third-party services.

3.7.4 Single User Account per Transaction

- Each user will be required to log in or register before proceeding with a transaction. Only one account can be associated with a single transaction to ensure tracking and personalized service.

3.7.5 Compliance with Data Privacy Laws

- The system will comply with relevant data privacy regulations, including GDPR and CCPA, ensuring that user data is collected, processed, and stored in compliance with legal requirements. Consent will be required from users before any personal data is processed, and users will have access to delete their data upon request.

3.7.6 Showroom and Ticket Management

- The system will automatically manage the capacity of each showroom based on the total number of seats available and the seats already reserved. It will prevent overbooking by ensuring that no more tickets are sold than there are seats available for a specific showtime.

3.7.7 Refund and Cancellation Policy

- The system will enforce a strict refund policy. Users can request a refund for a ticket up to 24 hours before the showtime. After this period, the system will no longer allow refunds. The system must prevent cancellations that violate this policy.

3.7.8 Scalability and Performance

- The system will be designed to handle at least 1000 concurrent users. It will be scalable, allowing for horizontal scaling of server resources when traffic increases.
- The database design will ensure efficient access to data, minimizing query times, especially during peak hours when many users might be purchasing tickets simultaneously.

3.7.9 Movie Management by Admin

- Admin users will be able to add, delete, and update movie details. Only one Super Admin will have the ability to remove movies, while other admins will be limited to adding and updating existing movies.

3.7.10 User Feedback and Ratings

- The system will allow users to rate movies and leave feedback. However, users will only be able to rate a movie after attending the show, ensuring that only verified viewers can provide feedback.

3.7.11 Role-Based Restrictions on Employee Access

- Employees will have restricted access to certain admin functionalities. For example, cashiers can only view user tickets, while movie managers can add or update movie showtimes but cannot delete movies. This ensures that only employees with the right authority can access certain parts of the system.

3.7.12 Backup and Disaster Recovery

- The system will have a disaster recovery plan that includes daily backups of the entire system database. In the event of a system failure or data loss, the system should be restorable from these backups within 60 minutes.

3.7.13 Session and Data Management

- Each user session will be stored temporarily and will expire automatically if no activity is detected within a 15-minute window. Additionally, once a user completes the purchase process, their shopping cart will be cleared automatically, and the session will be ended securely.

3.7.14 High Availability and Uptime

- The system must guarantee 99.98% uptime to ensure smooth and continuous access. Redundant systems and failover strategies must be in place to reduce downtime and ensure that the system can recover from unexpected failures.

3.7.15 Movie Viewing History

- Users will be able to access their viewing history through their account. This history will store details of movies they have watched, including dates and showtimes. The system will retain viewing history for one year before it is archived or deleted.

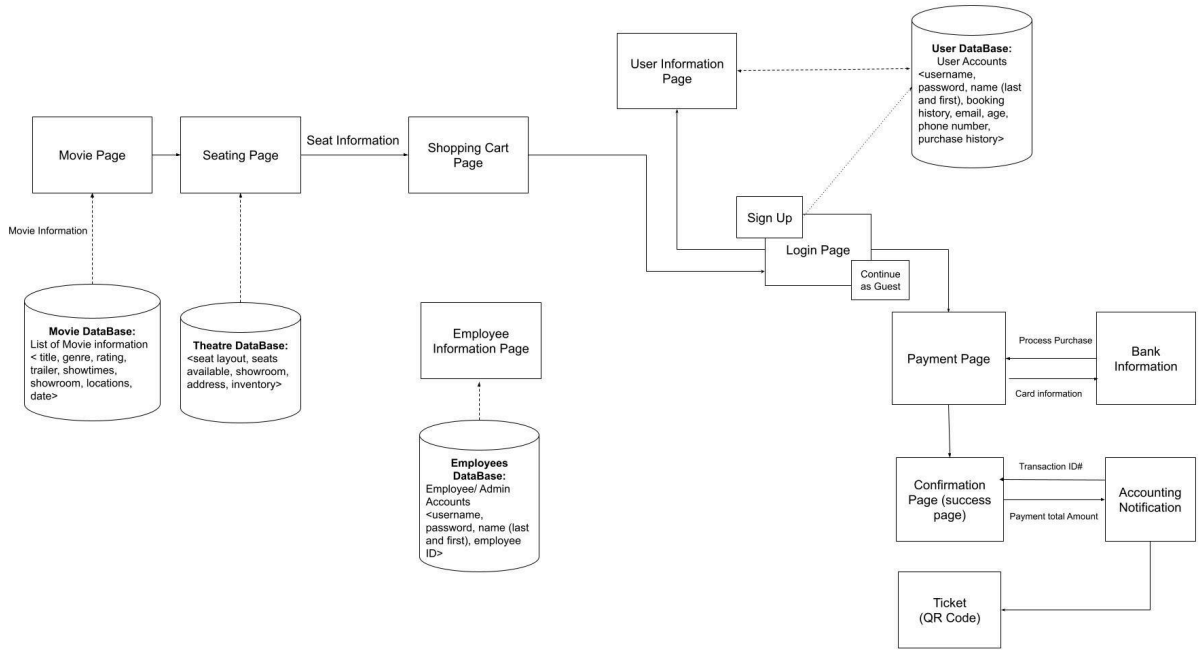
4. Software Design Specification

4.1 System Description

The movie ticketing system has been created to manage and streamline the process of booking tickets at Blossom Theaters. It allows the customers to easily book movie tickets, select their desired location, rate the movies, choose their seats, and ask for refunds. It also enables the employees and admins of the theater to manage the movies they will be showing in the showrooms and their time slots. Aside from this, it will also let them help customers in case of technical problems.

The software design is created to show how the classes of the movie ticketing system interact by using UML diagrams. Additionally, the software architecture diagram is made to show the flow of the system and what happens when a user starts to book a movie. It also displays how the database is accessed within the system.

4.2 Software Architecture Diagram



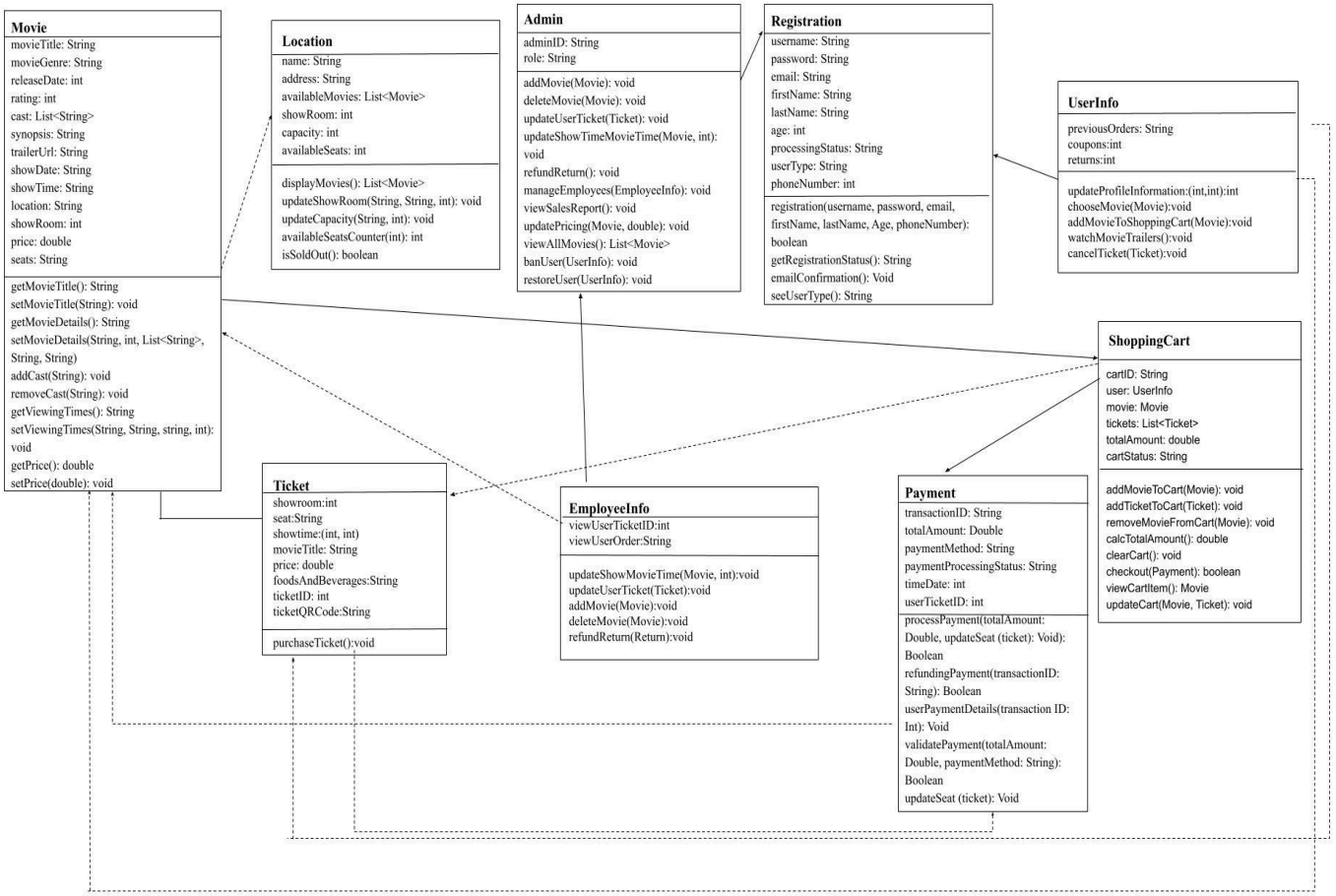
The movie theater ticketing system for Blossom Theaters is designed to make the purchase process easier for both employees and customers, enhancing the customer's experience. This will consist of several user-friendly features and databases to store movie information, employee/ admin information, and customer information.

The main actors in this system will consist of the customer user, admin, and employees (as well as any system they interact with). Users (or customers) will be able to look at movie details, showtimes, seats, and purchase tickets and or food.

This process is behind the movie page, where the user will choose the showtime for the movie they wish to watch. From here, they will be able to choose their seats, and this is where the seat reservation will be held. Next, the customer is led to the shopping cart page, which shows food and drink options for purchase. From this page, the user will be led to the payment page, where the user will be prompted to either log in, create an account, or continue the transaction as a guest. From here, the user will be prompted to enter their card information, and once the transaction is approved by both the POS system and the bank, then, the user is led to a confirmation page where they will receive a message pertaining to the ticket being sent to the user's email. This system will also inform the accounting team that a sale has been made for a movie ticket.

Interactions between the user and system allows the viewing of real time availability while the staff has control to manage show times and bookings through use of the admin interface. This system will effectively sell tickets, improve customer satisfaction through convenience, and ensure that data management is as effective as possible.

4.3 UML Class Diagram



The UML Class Diagram shows the relationships, dependencies, and interactions between the classes that exist in the movie ticketing system. The movie class is the foundational class in the system. Without the movie class most of the operations in the classes are not going to work. The ticket class and shopping cart class is associated with the movie class. This means that the shopping cart class and ticket class exists independently but uses the information from the movie class such as the title of the movie, the showtime, and the price of the movie. The movie class is dependent on the location class because the movie needs information from the location to use some of its operations, like assigning the showroom and setting the locations the movie is shown. The user info class and payment class is also dependent on the movie class. The user info class, employee info class and admin class all have a directed association with the registration class. This means that these classes are aware about the registration class but the registration class does not need to be aware of these classes to perform its operations.

4.3.1 Movie Class

The movie class contains all the key attributes and operations of the movies. Each movie shown in the theater will be represented as an instance of the movie class. The class will also serve as the blueprint for the admin staff as they enter information about the movie shown in the theater.

4.3.1.1 Attributes

- movieTitle: String
    - The attribute stores a string that represents the name of the movie.
- movieGenre: String
    - The attribute stores a string value that represents the genre of the movie. The genre can be horror, action, comedy, and documentary.
- releaseDate: int
    - The attribute stores an integer value that represents the year the movie was released.
- rating: int
    - The attribute stores an integer value that represents the rating of the movie out of a five star rating system.
- cast: List<String>
    - The attribute stores a list of string values representing the cast members of the movie.
- synopsis: String
    - The attribute stores a string value that represents a short summary of the movie.
- trailerUrl: String
    - The attribute stores a string value that represents the URL link to view the trailer of the movie.
- showDate: String
    - The attribute stores a string value representing the dates the movie will show on Blossom Theaters.
- showTime: String

- The attribute stores a string value that represents the times the movie will be showing on Blossom Theaters.

- location: String
    - The attribute stores a string value that represents which Blossom Theater location the movie will be playing.

- showRoom: int
    - The attribute stores an integer value that represents the movie room number the movie will be playing based on the selected location.

- price: double
    - The attribute stores a double value that represents the price of each movie ticket for the movie

- seats: String
    - The attribute stores a string value that tells the number of seats available for the movie based on the selected location and showroom.

4.3.1.2 Operations

- getMovieTitle(): String
    - The function is used to retrieve the title of the movie and returns it as a string.

- setMovieTitle(String): void
    - The function takes a parameter of data type string, which will be used to assign the title of the movie.

- getMovieDetails(): String
    - The function is used to retrieve the details of the movie and returns these as a string. The details that will be returned are the genre, release date, rating, cast, synopsis, and the URL link of the movie.

- setMovieDetails(String, int, List<String>, String, String)
    - The function takes a parameter of data type string representing the genre, an integer representing the release date, a list of strings representing the cast members, a string representing the show date, and a string representing the show time. The admin will use the parameters to update or add information about the selected movie.

- addCast(String): void

    - The function takes a parameter of data type string representing the name of the cast to be added to the movie information.

- removeCast(String): void

    - The function takes a parameter of data type string representing the name of the cast to be removed from the movie's cast members.

- getViewingTimes(): String

    - The function will return a string showing the times the selected movie will play in a specific location. It will be displayed using the 12-hour clock format.

- setViewingTimes(String, String, String, int): void

    - The function takes a parameter of data type String representing the show date, a string representing the show time, a string representing the location, and an integer representing the showroom number. The admin staff will use this to set the date, times, and location where a particular movie will be played.

- getPrice(): double

    - The function returns a double representing the price of the movie.

- setPrice(double): void

    - The function takes a parameter of data type double. It is used to set the price of the movie in US dollars.

4.3.2 Location Class

The location class contains the attributes and operations needed to set up the branch locations of Blossom Theaters. It will show how many showrooms are available for each location and how many seats are available for each showroom. This will also allow the admin staff to select which locations and showrooms a specific movie will be playing.

4.3.2.1 Attributes

- name: String

    - The attribute stores a string that represents the name of the location.

- address: String

    - The attribute stores a string that represents the address of the location.

- availableMovies: List<Movie>

    - The attribute stores a list of movies representing all the movies playing in that

location.

- showRoom: int
    - The attribute stores an integer representing the number of showrooms in the selected location.
- capacity: int
    - The attribute stores an integer that represents the capacity of the location.
- availableSeats: int
    - The attribute stores an integer that represents the number of available seats in each showroom of a specific location.

4.3.2.2 Operations

- displayMovies(): List<Movie>
    - The function returns a list of movies that are currently available in the selected location.
- updateShowRoom(String, String, int): void
    - The function takes a parameter of data type string representing the movie that will be playing in that location, a string representing the show time the movie will be played, and an integer indicating the showroom number. This function will update what movie will be played and when it will play in the indicated showroom number.
- updateCapacity(string, int): void
    - The function takes a parameter of a data type string representing the name of the location and an integer representing the number of people allowed in that location. It will be used to update the capacity of the location.
- availableSeatsCounter(int): int
    - The function takes a parameter of data type integer representing the available number of seats in the selected showroom in the location. It is also used to count down how many seats are still available to be purchased and prevent overbooking.
- isSoldOut(): boolean
    - The function returns a boolean and will tell the system if there are still seats available in the selected showroom. If it returns true, the system will allow the customer to book the seats. If it returns false, the system will prevent the customer from booking the seats and will display a pop out stating that all the tickets are

sold out.

4.3.3 Ticket Class

The ticket class will represent the ticket that the customer will have in which it would include chosen seat, movie title, price, and chosen foods and beverages if any by the customer. Not only that, but there will also be ticket ID along with the ticket QR code that are going to be part of the ticket

4.3.3.1 Attributes

- showroom: int
    - The showroom number would be a whole number that would be represented by an integer. The showroom number would be displayed on the ticket after being purchased by the customer.
- seat: String
    - The customer will be able to view their seats in which it is going to be a matter of characteristics made out of letters and integers as part of the seating system. This will include the seating row and column.
- showtime:(int, int)
    - The showtime will be displayed on the ticket by the hour and the minute. Therefore, the hour and the minute will be used as integer parameters.
- movieTitle: String
    - The movie title will be a string because it is going to be the name of the movie. This will be the way that the movie will be recognized and this will be displayed on the ticket to know which movie the customer will be watching.
- price: double
    - The price of everything that will be purchased such as the movie along with foods and beverages will be displayed on the movie ticket. The price will be used as a double so that it can hold prices as decimals.
- foodsAndBeverages: String
    - The foods and beverages will be used as a string because it will hold the name of the foods and beverages. The foods and beverages will be stored in the QR code that is part of the purchases made by the customer.
- ticketID: int
    - The ticket ID will be given to the customer as part of the ticket when it would be

purchased to differentiate each ticket from another one and it will be a one time used type of ticket. The ticket ID will be represented as integers to hold the whole number values.

- ticketQRCode:String
    - The ticket QR code will be used as a string to hold the name of the QR code that will be a one time use. It will include many letters and numbers as part of the link that goes with the QR Code, which is why it is going to be a string.

4.3.3.2 Operations

- purchaseTicket():void
    - The purchaseTicket() method will be responsible for the entire purchasing process of the ticket. It could be using the shopping cart where foods and beverages, seats, showtime and the movie would be stored or it could be purchased after choosing the movie. This method will not make any returns occur when the purchasing happens. However, this method will reserve the ticket for the customer that will differentiate each ticket one from another by giving the customer things like ticket ID and QR code that will be sent to the customer after the purchase happens.

4.3.4 UserInfo Class

The user information class will be about the customer in which this is where the information that would be accessible and operated by the customer. This means that they will have the ability to utilize the site in regards to their previous orders along with the coupons and any returns that the customer wants to make. There will also be operations that the customer will have the ability to do in which this include things like choosing a movie, adding a movie to their shopping cart, watching a movie trailer to know what the movie is about, and canceling their ticket.

4.3.4.1 Attributes

- previousOrders: String
    - The customer’s previous orders would be represented by a string, which is the name of their purchased items. This includes things like seat, showroom, movie title, ticket ID, ticket QR code, showtime, price, and food and beverages.
- coupons:int
    - The customer will have coupons after they make any purchases. Those coupons will be given in the whole number of dollars form of integers.
- returns:int
    - The customers will also have returns available to them in the case that they want to return their purchases. The returns would be used by integers, so everything is

done in the whole number form of items, so this means that a customer can return whatever number of items that they want to return.

4.3.4.2 Operations

- updateProfileInformation:(int,int):int
    - This operation is a method that gives the customer the ability to update their profile information. This method is done by replacing the old integer value with the new integer in which this updates it to the int that it was replaced by, so then it returns an integer value.
- chooseMovie(Movie):void
    - This operation happens when the customer chooses a movie in which this method takes care of the process of what happens when the customer gets to choose the movie. This method of choosing the movie happens by having the customer choose the movie they want and then the system would grab the movie variable as an input, but return nothing since it would take the user's input of the movie selection.
- addMovieToShoppingCart(Movie):void
    - This method is made from the selection made by the customer by adding a movie to the shopping cart. The way that this method works is by having the customer add a movie to their shopping cart in which the movie variable would be grabbed from the movie selection as an input made by the user and so this method would not return anything.
- watchMovieTrailers():void
    - This operation is made for the customer to have the ability to watch the movie trailer to get to know what the movie is about in which this would be accessible on the site for the customer. This is a method that does not need to have any input variables or anything that should be returned.
- cancelTicket(Ticket):void
    - This method is when the customer wants to cancel their ticket. This means that this method would have the customer select to cancel the ticket and then receive the ticket as an input variable from the user, but it would not return value and the ticket would be canceled.

4.3.5 EmployeeInfo Class

The employee information class is responsible for all the things that the employees have the ability to utilize and operate regarding the site update information. This class is also about what the user's ticket ID is along with their order just in case there would be a problem in which this

information would be used for reference and acceptance as well. The operations that the employees can do is update the show time of the movie and the user's ticket along with adding and deleting the movies that are on the site. The employees would also be able to validate the returns and accept refunding making sure that the ticket was not already used or damaged.

4.3.5.1 Attributes

- viewUserTicketID:int
    - The employees will have the ability to see the ticket ID number that the user has in which they receive this ticket ID when they purchase a ticket. This method is responsible for validating the ticket ID number, which is based on integers when being viewed.
- viewUserOrder:String
    - The employees would have the ability to look up and see the customer's order, which would be in a string because it is going to have a unique name with letters and numbers that differentiates it from other orders. The employees would be able to view this customer's order just in the case that there is a problem in the customer's order and when the customer is entering the theater to make sure that they have everything they have ordered.

4.3.5.2 Operations

- updateShowMovieTime(Movie, int):void
    - This method will allow the employees to update the show time of the movies that are on the website. This means that this method would work by the parameter of taking the movie name as an input along with the new time that is in an integer form with nothing to be returned.
- updateUserTicket(Ticket):void
    - The employee would have the ability to update the user's ticket in the case that the customer wanted anything to be updated or in the case that there is anything that should be updated regarding the showtime, showroom, movie or food and beverages availability, etc. The way that this method operates is by receiving the ticket as an input by the employees and any updates can happen from there without the need of having the function update anything.
- addMovie(Movie):void
    - The employees would have the ability to add movies to the website. This means that the movie that the employee wants to add would be received as an input with nothing to be returned.
- deleteMovie(Movie):void

- The employees would have the ability to delete any movie that is on the website in which this method would be used to take care of that operation. The way that this works is by receiving the movie title that the employee wants to be deleted and then it would be deleted with nothing that would be returned.

- refundReturn(Return):void

    - This method is made to take care of the ticket return process when the customer wants to be refunded in which it is made to validate if the ticket was not previously used or is damaged before the customer would get back their refund. The way that this method works is by receiving the returned item as an input and then once the validation happens then there would be nothing to be needed to return to our system and the customer would have their money refunded to them.

4.3.6 Payment Class

The payment class will be responsible for checking payment details and information. This class will take user input regarding card information, making sure the information is properly formatted and failing if the information is incorrect. Once the customer enters their card information out on the checkout page the card details will be encrypted and sent to the respective bank. Once the transaction is approved and verified then the seats will be reserved and the seating chart is updated to properly reflect the user's seat reservation (making them non-choosable).

4.3.6.1 Attributes

- paymentID:

    - A unique number for each transaction that helps track and reference specific transactions.

totalAmount:

    - The total Amount that is added up in the shopping cart. This takes the price of the tickets and any possible food (and tax).

- paymentMethod:

    - This is the payment method that the user will use, this can include options like credit/debit cards.

- paymentProcessingStatus:

    - This will be the sign on whether or not the payment has been processed or not. This will store the possible values of "complete", "pending", "failed" or "refunded".

- userTicker:

    - Once transaction is complete then it will go through the process of

4.3.6.2 Operations

- processPayment(totalAmount, paymentMethod, updateSeat): Boolean

    - This operation will start the processing of the payment and will validate the payment information. Processing the transaction and returning true if the payment was successful, else it returns false. This will also check if the seat is still available to reserve, if not then the payment will fail and the user will be prompted to choose different seats.

- refundingPayment(transactionID: Integer): Boolean

    - Operation will handle the refunding process for, verifying if the requirements for a refund have been met and will process the refund accordingly.

- userPaymentDetails(transactionID: Integer): String

    - This operation will retrieve and format the payment details of the transaction, this will consist of: paymentID, pamentMethod and paymentProcessingStatus. Since we will not store payment information of users, this function will NOT store this information.

- validatePayment(totalAmount: Float, paymentMethod: String): Boolean

    - This operation will validate the payment information before processing and continuing. It will check if the payment method and amount is valid and will return true if valid. Otherwise it will return as false

- updateSeat (ticket): Void

    - This operation will update the seat that was chosen for this transaction by the user. This method will not return a value but will set reservations for seats in the system (making them not able to be purchased by other later users).

4.3.7 Registering Class

The class is designed to provide the functionality of account creations, validating login information and distinguishing who is a guest and who is an account holder/user. It has the ability to regulate or facilitate user accounts, enabling both guests and customers that are already registered to interact with the system's database. This class will streamline the registration process, handle logins and provide a point of access to the payment page.

4.3.7.1 Attributes

- Username: string

    - This will be a unique username that is chosen by the user for their account, this is a required input for registered users but not for guests.

- Password: string
    - This will be the password that will be stored matching the username, this is made for account security. This is required for registered users but not for guest users. This password must meet the requirement of: Has to obtain at least 8 characters, one uppercase, one lowercase, 1 number and 1 special character.
- Email: string
    - The email is one that the user owns, and has not been used before for another user's account. This is what will be used to communicate to customers (and verification). This is required for registered users but not for guest users
- firstName: string
    - First name of the user, Necessary for all users, guests or returning (This is a one time input for registered users) This is what will be printed on the ticket along with the QR code.
- lastName: string
    - Last name of the user, a requirement for registered users but not for guest users. This is a one time input for registered users and will be stored in the customer profile.
- Age: integer
    - This will store the age of the user. This will make it so that movie suggestions that align with their age
- processingStatus: string
    - This will store if the registration process was successful or not
- userType: String
    - This will be how guests and registered users are recognised. This will make it so the system will follow a set of rules for guest users and for registered users.
- phoneNumber: integer
    - This value will be stored in the user profile and enable registered users to receive text message promotions.

4.3.7.2 Operations

- registration(Username, Password, Email, firstName, lastName, Age, phoneNumber): Boolean

    - This operation will handle the registration process for making a new account. This will be in charge of making sure that all fields are formatted correctly and failing if there are any fields filled out wrong. This operation will return true if everything is filled in correctly and if registration is successful. If it returns false then that means that the operation failed.

- validateEmail(email): Boolean

    - This operation will check if the email that was input is formatted correctly (ie. has '@' symbol, uses gmail, yahoo ect.) If this returns true then it will determine that the email is valid and continue. This will also check if the email is being used. If this returns false then the user is prompted to re enter their email.

- validatePassword(password): boolean

    - This operation will validate the password by checking if the criteria (Has to obtain at least 8 characters, one uppercase, one lowercase, 1 number and 1 special character) has been met. If this returns false then that means that the criteria has not been met . If this returns true then that means the user input has satisfied all criteria.

- getRegistrationStatus(): string

    - This operation will check the status of the users current registration process and a message on whether or not the registration was successful.

- emailConfirmation(): Void

    - This operation will send out an email confirmation to the user once the registration process is successful, this will confirm that the email actually belongs to the customer. This will not return anything but rather send out an email.

- setUserType(): String

    - This operation will use the user type once the registration is processed. This will set registered users as "registered", else it will assign the user as a guest.

4.3.8 Admin Class

The Admin class represents the administrator of the Blossom theater system. The administrator has special privileges to manage the overall system, such as updating movie schedules, managing user tickets, handling refunds, and interacting with both users and employees. The Admin class is responsible for overseeing the core functionalities of the system and ensuring that it operates smoothly.

4.3.8.1 Attributes

- adminID: String

  - A unique identifier for the Admin user in the system.

- role: String
  - defines the role or level of access the admin has within the system.

4.3.8.2 Operations

- addMovie(Movie): void
  - This function allows the admin to add a new movie to the system. The admin provides the movie, which contains all relevant details such as title, genre, release date, cast, showtimes, and price. Once added, the movie becomes available for customers to select and purchase tickets.

- deleteMovie(Movie): void
  - This function allows the admin to remove a movie from the system. Once removed, the movie will no longer be available for users to buy tickets, but any already purchased tickets for that movie should still be honored.

- updateUserTicket(Ticket): void
  - This function allows the admin to update a user's ticket. The admin can modify details such as the seat number, showtime, or movie title. This might be necessary if there was an error in the original booking or if the user requested a change.

- updateShowTimeMovieTime(Movie, int): void
  - The admin uses this function to modify the showtime for a specific movie. It also updates the viewing schedule of a movie, which could be necessary due to changes in the theater's schedule or special events.

- refundReturn(): void
  - This function processes refunds for users who wish to cancel their tickets. The admin can approve or deny the request based on the Blossom Theaters' refund policies.

- manageEmployees(EmployeeInfo): void
  - The admin uses this function to manage employee information, such as viewing their assigned tasks, updating their details, or even adding or removing employees from the system.

- viewSalesReport(): void
  - This function generates and displays a sales report, providing the admin with a summary of ticket sales, revenue, and possibly trends based on user purchasing behaviors.

- updatePricing(Movie, double): void
    - The admin can use this function to change the ticket price for a specific movie, adjusting prices based on demand, special events, or promotional periods.
- viewAllMovies(): List<Movie>
    - This function allows the admin to retrieve and view the entire list of movies currently available in the system. This function returns a list of movies, which represent the movies that are either currently playing or scheduled to be shown at the theater.
- banUser(UserInfo): void
    - This function allows the admin to ban a user from the system. This could be done for various reasons such as violation of terms, fraudulent activities, or repeated refund requests. Once banned, the user will no longer be able to access the system.
- restoreUser(UserInfo): void
    - The admin can use this function to restore a banned user, allowing them to regain access to the system. This could be useful in cases where users were banned temporarily and are now allowed to resume activity.

4.3.9 ShoppingCart Class

The ShoppingCart class is responsible for temporarily holding the user's selected movies and tickets before finalizing the purchase through the payment system. The cart stores essential information about the selected movies, associated tickets, and the total cost. It also manages actions such as adding, removing, or updating items and proceeding to checkout.

4.3.9.1 Attributes

- cartID: String
    - A unique identifier for the shopping cart instance.
- user: UserInfo
    - Represents the user who is currently interacting with the shopping cart.
- movie: Movie
    - Stores the movie that the user has added to their shopping cart.
- tickets: List<Ticket>
    - Represents the tickets associated with the movie in cart.
- totalAmount: double

    - Holds the total price of all the tickets including other fees in the shopping cart

- cartStatus: String
    - The current status of the shopping cart. Statuses are “Pending”, “Completed”, or “Abandoned.”

4.3.9.2 Operations

- addMovieToCart(Movie): void
    - This function allows a customer to add a movie to the shopping cart. If a movie is already in the cart, the function will either replace the existing movie or prompt the user to remove the current movie before adding a new one.
- addTicketToCart(Ticket): void
    - This function adds a ticket to the ticket lists. The ticket holds details about the showtime, seat selection, and price for a particular movie.
- removeMovieFromCart(Movie): void
    - This function removes a movie and any associated tickets from the cart. The user might decide to no longer watch a particular movie and would want to delete it from their cart.
- calcTotalAmount(): double
    - This function calculates the total price of all the tickets currently in the shopping cart. It adds up the price of each ticket in the ticket lists and returns the total amount.
- clearCart(): void
    - This function clears all items from the shopping cart after a successful checkout or if the user decides to abandon the cart. It empties the movie and ticket lists and resets the totalAmount.
- checkout(Payment): boolean
    - This function finalizes the shopping cart by processing the user's payment. It calls the processPayment() function from the Payment class to handle the financial transaction. If the payment is successful, it clears the cart and updates the cardStatus to “Completed.”
- viewCartItem(): Movie
    - This function allows the user to view the movie they have added to the shopping cart. It returns the movie currently in the cart.
- updateCart(Movie, Ticket): void

- This function allows the user to update a movie or ticket selection in the cart. For example, the user might change the showtime or seat selection.

4.4 Development Plan and Timeline

The table shows the timeline each task was assigned, created, finished, and who was responsible for it.

List of Tasks	Estimated Timeline	Team Member Responsible
3.7 Design Constraints	10/06/24 - 10/08/24	Judge Dred Banal
4.1 System Description	10/08/24	Abigail Dupaya
4.2 Software Architecture Diagram and Description	10/06/24	Rita Yousif, Jesyl Zendejas, Abigail Dupaya
4.3 UML Class Diagram	10/06/24	Rita Yousif, Jesyl Zendejas, Abigail Dupaya, Judge Dred Banal
4.3.1 Movie Class	09/30/24 - 10/06/24	Abigail Dupaya
4.3.2 Location Class	09/30/24 - 10/06/24	Abigail Dupaya
4.3.3 Ticket Class	09/30/24 - 10/06/24	Rita Yousif
4.3.4 UserInfo Class	09/30/24 - 10/06/24	Rita Yousif
4.3.5 EmployeeInfo Class	09/30/24 - 10/06/24	Rita Yousif
4.3.6 Payment Class D	09/30/24 - 10/06/24	Jesyl Zendejas
4.3.7 Registering Class	09/30/24 - 10/06/24	Jesyl Zendejas
4.3.8 Admin Class	09/30/24 - 10/06/24	Judge Dred Banal
4.3.9 ShoppingCart Class	09/30/24 - 10/06/24	Judge Dred Banal

5. Test Plan

5.1 Test Plan

5.1.1 Introduction

Purpose: The purpose of this test is to outline the testing process for the Blossom Theaters system to ensure its functionality, usability, performance, and security across different modules.

Scope: The test plan covers functional, performance, and security testing for key features such as movie search, booking tickets, shopping cart, user registration, payment processing, and error handling.

Objectives: Ensure that users can smoothly search for movies, book tickets, manage their shopping cart, and make payments while receiving appropriate error messages when needed.

5.1.2 Test Strategy

Types of Testing

- **Functional Testing**: Verifying the correct functionality of each feature.
- **Usability Testing**: Ensure the website is easy to navigate and user-friendly.
- **Performance Testing**: Test how the website performs under different load conditions.
- **Security Testing**: Test the website for vulnerabilities in the login, registration, and payment processing modules.
- **Compatibility Testing**: Verify the website across different browsers and devices such as desktop, tablet, and mobile.

Testing Tools

- **Functional**: Selenium, Postman (for API testing)
- **Performance**: JMeter, LoadRunner
- **Security**: OWASP ZAP, Burp Suite
- **Browser Compatibility**: BrowserStack

5.1.3 Test Scope

In-Scope

- Movie Search (misspelling, filters, and expired Showtimes)
- User Authentication (login and registration)

- Shopping Cart(Limits, Capacity)
- Payment Processing
- Error Handling

Out-of-Scope

- Integration with third-party services like social media.

5.1.4 Test Environment

Hardware: Desktop PCs, Mobile devices, Tablets.

Operating Systems: Windows 10, macOS, Android, iOS.

Browsers: Chrome (latest version), Firefox (latest version), Safari, Edge.

Test Data: Test accounts, sample movies, mock payment details (for invalid and valid card information).

5.1.5 Test Cases

Case 1: Misspelling Search

- **Component**: Movie_Search
- **Purpose**:
    - Verify that the system handles misspelled movie searches by showing an error message or a suggestion to correct the spelling.
- **Pre-requisites**:
    - The movie search module is fully functional, and a database of available movies is set up.
- **Test Steps**:
    1. Open the browser and enter https://blossomtheaters.com.
    2. Once the homepage loads, locate the search bar.
    3. Enter a misspelled movie name (e.g., "Spidrman").
    4. Press Enter.
- **Expected Results**:

    - The system should return an error message such as "No results found" or suggest possible correct movie titles.

- **Failure Cases**:
    1. The system returns irrelevant movies instead of an error or correction.
    2. No error message is displayed, and the system does nothing.
- **Rationale:**
    - Users often make spelling mistakes, and the system should provide assistance, such as suggestions or corrections, to improve the user experience.

Case 2: The Search Bar Left Blank

- **Component**: Movie_Search
- **Purpose**:
    - Verify that the system prevents users from submitting a blank search query.
- **Pre-requisites**:
    - Movie search functionality should be available.
- **Test Steps**:
    1. Open https://blossomtheaters.com.
    2. Click on the search bar.
    3. Leave the search bar blank or input only spaces.
    4. Press Enter.
- **Expected Results**:
    - The system should prevent the search from being executed and display an error message like “Please enter a search term.”
- **Failure Cases**:
    1. The system executes a search despite the blank search query.
    2. No message is displayed, and the system does not give feedback.
- **Rationale:**

- A blank search query leads to unnecessary system load and poor user experience, so this functionality helps prevent invalid searches.

Case 3: Password or Username Search

- **Component:** Login_Module
- **Purpose:**
  - Verify that an incorrect username or password results in an appropriate error message during login.
- **Pre-requisites:**
  - User login functionality must be operational, and a test account with known credentials should be available.
- **Test Steps:**
  1. Open the browser and navigate to https://blossomtheaters.com.
  2. Go to the login page.
  3. Enter incorrect username and password.
  4. Press Enter.
- **Expected Results:**
  - An error message should be displayed, such as "Invalid username or password."
- **Failure Cases:**
  1. The system logs in the user despite incorrect credentials.
  2. No error message is shown after an invalid login attempt.
- **Rationale:**
  - Proper error handling during login is crucial for maintaining security and preventing unauthorized access.

Case 4: Expired Showtime Search

- **Component**: Movie_Page_Module
- **Purpose**:
  - Verify that the system does not allow booking a showtime that has already passed.

- **Pre-requisites**:
    - Movie listings and showtimes should be set up with expired entries.
- **Test Steps**:
    1. Open the browser and go to https://blossomtheaters.com.
    2. Navigate to the movie page.
    3. Select a movie and attempt to book a showtime that has already expired.
    4. Press Enter.
- **Expected Results**:
    - The system should display an error message like “This showtime is no longer available.”
- **Failure Cases**:
    1. The system allows the user to book an expired showtime.
    2. No message is displayed, and the system proceeds without issue.
- **Rationale**:
    - To prevent booking confusion, the system should ensure users can only select valid and future showtimes.

Case 5: Purchase out of Items

- **Component**: Shopping_Cart_Module
- **Purpose**:
    - Verify that the system properly handles stock limitations for food, drinks, and tickets. If a user attempts to purchase more items than available, an error message should be displayed, preventing the transaction from proceeding
- **Pre-requisites**:
    - The system must have stock data configured, specifying limits for each item (e.g., maximum available stock for popcorn, drinks, and tickets).
    - A user should have valid credentials and be able to log in.
- **Test Steps**:

1. Open a web browser and enter the URL: https://blossomtheaters.com in the browser's URL bar and press Enter.
2. Log in with valid user credentials.
3. Navigate to the shopping cart page.
4. Add an excessive number of items to the cart
5. Click on the "Continue to Payment" button.

- **Expected Results**:
    - The system should prevent the user from proceeding to the payment page when the number of selected items exceeds the available stock.
    - An appropriate error message should be displayed (e.g., "The selected quantity exceeds available stock for popcorn" or "You cannot purchase more than 10 tickets").
- **Failure Cases**:
    1. The system allows the user to proceed to the payment page without verifying the available stock, resulting in over-purchase.
    2. The system shows an incorrect error message that doesn't accurately describe the issue.
    3. The system doesn't update stock in real-time, showing outdated stock levels during checkout.
- **Rationale**:
    - Ensuring that the system can handle stock management effectively is critical for preventing issues with overselling and maintaining an accurate inventory.

Case 6: User Capacity

- **Component**: Admin_Module
- **Purpose**:
    - Verify that the system redirects users to a waiting page when the website is at maximum capacity.
- **Pre-requisites**:
    - Admin controls must be in place to monitor and limit user capacity.

- **Test Steps**:
    1. Open the browser and enter https://blossomtheaters.com.
    2. Simulate or generate high traffic.
    3. Observe whether the system redirects users to a waiting page once the capacity limit is reached.
- **Expected Results**:
    - Users exceeding the limit should be redirected to a message like “The site is currently experiencing high traffic. Please try again later.”
- **Failure Cases**:
    1. The system allows too many users, causing performance issues.
    2. No message or redirection occurs when the capacity is exceeded.
- **Rationale**:
    - Managing website load is essential for maintaining a smooth user experience, especially during peak times.

Case 7: Error Display in Registration

- **Component**: Registration_Module
- **Purpose**:
    - Verify that the system displays appropriate error messages when incorrect or incomplete information is entered during registration.
- **Pre-requisites**:
    - Registration functionality should be working.
    - Test data includes invalid input (e.g., invalid email format).
- **Test Steps**:
    1. Navigate to https://blossomtheaters.com.
    2. Click on the registration page.
    3. Enter invalid information, such as a missing password character or an incorrect email format.

4. Press the submit button.

- **Expected Results**:
    - An error message should be displayed, prompting the user to correct their input.
- **Failure Cases**:
    1. The system allows invalid information and completes the registration.
    2. No error message is displayed.
- **Rationale**:
    - Ensuring proper data validation during registration is crucial for maintaining data integrity and user satisfaction.

Case 8: Payment Validation

- **Component**: Confirmation_Page_Module
- **Purpose**:
    - Verify that incorrect or invalid payment information results in an appropriate error message and that the system does not proceed with the payment.
- **Pre-requisites**:
    - Payment processing functionality must be integrated and functional.
    - A valid test account for logging in should be available.
    - The system should be configured with both valid and invalid payment test cases.
- **Test Steps**:
    1. Open the browser and navigate to https://blossomtheaters.com.
    2. Log in with valid user credentials.
    3. Select a movie and showtime.
    4. Choose seats and add them to the cart.
    5. Go to the shopping cart and confirm the selected items.
    6. Proceed to the payment page.
    7. Enter incorrect payment details

8. Press the “Confirm Payment” button.

- **Expected Results**:
    - The system should display an error message like "Invalid payment details" or "Payment could not be processed due to incorrect information."
    - The payment should not be processed, and the user should remain on the payment page for correction.
- **Failure Cases**:
    1. The system allows the payment to proceed despite incorrect card details.
    2. No error message is displayed, leaving the user confused about the issue.
- **Rationale**:
    - Payment validation is critical to ensure secure transactions and prevent fraudulent activities. Users must be notified if incorrect payment information is entered so they can correct the details before proceeding with the transaction.

Case 9: FilledShoppingCart

- **Component**: Shopping_Cart_Module
- **Purpose**:
    - Verify that the system prevents users from adding more items once the shopping cart reaches its maximum capacity.
- **Pre-requisites**:
    - Shopping cart functionality must be operational with predefined maximum limits for items.
- **Test Steps**:
    1. Open https://blossomtheaters.com and log in.
    2. Add items to the shopping cart until the maximum capacity is reached.
    3. Attempt to add one more item beyond the capacity limit.
- **Expected Results**:
    - The system should prevent further items from being added and display a message such as “Cart limit reached.”

- **Failure Cases**:
    1. The system allows more items than the maximum capacity.
    2. No error message is shown when the cart is full.
- **Rationale**:
    - Limiting the shopping cart capacity helps manage stock and prevents errors during checkout.

Case 10: FilteringSearches

- **Component**: Movie_Search
- **Purpose**:
    - Verify that the movie search function filters results based on selected criteria like genre, rating, and date.
- **Pre-requisites**:
    - Filtering options should be implemented in the movie search feature.
- **Test Steps**:
    1. Navigate to https://blossomtheaters.com.
    2. Enter a search term in the search bar.
    3. Apply filters such as genre (Action, Drama), rating (PG-13), and date (current month).
    4. Press Enter to view filtered results.
- **Expected Results:**
    - The system should display search results that match the selected filters.
- **Failure Cases:**
    1. The system ignores the filters and shows unfiltered results.
    2. No search results are returned, despite matching filters.
- **Rationale**:
    - Filters improve search accuracy and help users find movies according to their preferences.

Case 11: TestAvailableSeats

- **Component**: Location_Module
- **Purpose**:
    - Verify that the system correctly tracks the number of available seats for each showtime and updates it in real-time after ticket purchases.
- **Pre-requisites**:
    - Seat tracking functionality must be working.
- **Test Steps:**
    1. Navigate to https://blossomtheaters.com.
    2. Select a movie and showtime.
    3. Choose and add 5 tickets to the cart.
    4. Complete the purchase and verify that the number of available seats has been updated.
- **Expected Results**:
    - The available seat count should decrease by 5 after the purchase.
- **Failure Cases:**
    1. The system does not update the available seat count.
    2. The seat count decreases incorrectly.
- **Rationale**:
    - Keeping track of available seats is critical for managing showtime capacity and preventing overbooking.

5.1.6 Exit Criteria

- All test cases must be executed.
- No critical bugs should remain unresolved.
- 100% of high-priority test cases must pass.
- 90% of medium-priority test cases must pass.

5.1.7 Test Deliverables

- **Test Case Document**:
    - A document detailing all test cases.
- **Bug Reports**:
    - Reports of any defects, with severity and priority levels.
- **Test Execution Reports**:
    - Daily reports on the progress of test case execution.
- **Final Test Report**:
    - A summary of the testing activities, outcomes, and key findings.

5.2 Test Case

Test Case Template									
TestCaseId	**Component**	**Priority**	**Description/Test Summary**	**Pre-requisites**	**Test Steps**	**Expected Result**	**Actual Result**	**Status**	**Test Executed By**
MisspellingSearch	Movie_Search	P9	Verify the movie search that is being made to compare with movies that are available and presses enter. When incorrect spelling occurs for the movie title, then an error message should be displayed.	A search must be made in the movie page first before verified for misspelling.	1. Enter this URL https://blossomtheaters.com and press enter. 2. Once the site comes up, it is going to show the movie page along with a search bar. 3. Write the movie name desired by the user. 4. Press enter.	There would be an error message to guide that user to check the spelling of their entries for the movie title.	There would be an error message displayed, which is “Please check spelling”	Pass	Rita
SearchBarLeftBlank	Movie_Search	P10	Verifies that the search bar will prevent the user from searching anything if it is left blank.	User is on the movie page.	1. Write the url - https://blossomtheaters.com in the browser’s URL bar and press enter. 2. Once in the movie page,	Enter button is grayed out and is disabled.	Enter button is grayed out and cannot be pushed, signaling it is disabled.	Pass	Abigail

					3. Write blank spaces on the search bar or leave it blank.				
PasswordOrUsernameSearch	Login_Module	P1	Verify the password and username with the login search to make sure that the exact password and username are provided and press enter. When incorrect information is provided, then an error message should be displayed.	The login page must be launched first and then enter the password and username before verifying if the entries are valid.	1. Enter this URL https://blossomtheaters.com in the browser's URL bar and press enter. 2. When the login page is reached, enter the username and password for the account. 3. Press enter afterwards.	There would be an error message that would be displayed to have the user check their password and username because it was an invalid entry and to make a re-entry of the information.	There would be an error message displayed, which is "Please check for incorrect password or username and try again"	Pass	Rita
ExpiredShowTImeSearch	Movie_Page_Module	P5	Verify the movie showtime to the movie page to make sure the chosen movie time is available and press enter. When movie time has expired, then an error message would be	The movie page must be launched first and then the showtimes would be chosen to verify the validity of the entries like if the movie	1. Enter this URL https://blossomtheaters.com in the browser's URL bar and press enter. 2. Go on the movie page where it will	There would be an error message that would be displayed for the chosen showtime to the user to notify them that the showtime they desire is no longer available since it has	There would be an error message displayed, "Please check other showtimes available since this movie showtime has already expired"	Pass	Rita

			displayed.	showtime has expired or not.	contain the showtimes of the movie. 3. The customer will enter the movie name that they want to watch. 4. The user will choose the showtime of the movie that they want to watch. 5. Press enter afterwards.	expired.			
PurchaseOutOfItems	Shopping_Cart_ Module	P3	In charge of making sure we have items in stock, if either case not satisfied an error message will display	User has added items that are out of stock into the shopping cart and clicks the continue to payment button.	1. User enters https://blossomtheaters.com in URL Bar 2. User chooses a movie, showtime (in this case random) 3. User chooses their seats from seating page 4. Users will continue to the shopping cart page, where they are able	Website will not allow user to continue to confirmation screen and will reload the shopping cart	Will display an error message of “(Item) is out of stock or exceeds limits, please remove item and try again”	Pass	Jesyl

					to add items desired (in this case said item is out of stock) 5. User will press the continue to payment page				
UserCapacity	Admin_Module	P0	This function will handle if there are too many users, directing them to a page with a message if needed.	User is attempting to enter website when busy	1. User enters https://blossomtheaters.com in URL Bar, and press enter.	Will block out any other users from entering onto a full website (causing issues)	Will display a page with a message saying “Sorry! It seems we are full. Please reload page and try again”	Pass	Jesyl
ErrorDisplay	Registration_Module	P2	Verify that when a user registers, the fields such as email, password, and phone number meet the minimum criteria.	User inputs invalid email, phone number, or insufficient password character count.	1. Write the url - https://blossomtheaters.com in the browser’s URL bar and press enter. 2. On the homepage, press the register button. 3. Users will be taken to the register page. 4. User inputs incorrect email format, missing password character, or	A pop up will appear stating, “Invalid input! Check if fields are properly filled. “	Error message popped up, stating, “Invalid input! Check if fields are properly filled.”	Pass	Abigail

					invalid phone number. 5. Click the submit button.				
PaymentValidation	Conformation_Page_Module	P4	In charge of making sure payments are valid, this confirms all the information being input by the user is valid.	A payment method has been input and result/ process failed	1. User enters https://blossomtheaters.com in URL Bar 2. User chooses a movie, showtime (in this case random) 3. User chooses their seats from seating page 4. User is then lead to shopping cart to confirm their items 5. User is lead to payment page, this is where user will make mistake (known as putting in wrong card information) 6. Users press the "confirm payment"	Will not allow the user to continue to the confirmation/ ticket page. Will reload payment page with blank fields and an error message	Will reload the page with blank fields, prompting the user to enter their payment information again, also displaying the message of "Sorry, payment processing failed, please re-enter your information and try again!"	Pass	Jesyl

					button.				
FilledShoppingCart	Shopping_Cart_ Module	P7	Verify that when the shopping cart reaches its maximum capacity, the system prevents further items from being added and displays an appropriate error message.	Users must be logged in and have access to a shopping cart.	1. Open https://blossomtheaters.com and press enter. 2. Login with valid credentials and navigate to the shopping platform. 3. Add items to the shopping cart one by one until it reaches its maximum allowed capacity. 4. Attempt to add one more item beyond the maximum capacity.	The system should display an error message indicating that the shopping cart has reached its limit and no further items can be added.	An error message is displayed: “Shopping cart is full. Cannot add more items.”	Pass	Judge
FilteringSearches	Movie_Search	P8	Verify that the movie search function correctly filters search results based on user-selected filters like genre, rating, or date.	The movie search page should be accessible, and the filtering options should be functional.	1. Open the browser https://blossomtheaters.com and navigate to the movie search page. 2. Enter the movie search	Only movies that match the selected filter(s) should be displayed in the search results.	Search results are displayed with only the movies that match the selected filter. The message displayed: "Filtered search results displayed for genre: Action, Drama,	Pass	Judge

					module and type in a general search term. 3. Apply a filter and select Action, Drama, Romance. 4. Press the search button to view filtered results.		Romance"		
TestAvaiableSeats	Location_Module	P6	Checks that availableSeats() function which is in charge of tracking the number of available seats is working properly. When a user books a number of tickets, this number should be deducted to the number of available seats and the quantity of available seats in the movie page should be updated.	Users need to book at least one movie ticket.	1. Write https://blossomtheaters.com in the URL bar and press enter. 2. Once in the movie page, select a movie. 3. Ente 5 in the quantity. 4. Add them to your cart. 5. Go to your cart. 6. Press checkout.	Available seats are updated with the correct quantity. Before booking a ticket, the quantity of available seats was 65. After booking 5 tickets, the quantity of available seats is updated to 55.	Available seats displayed the updated quantity of seats, 1 milliseconds after booking the ticket.	Pass	Abigail

6. Data Management

6.1 SWA data management Diagram

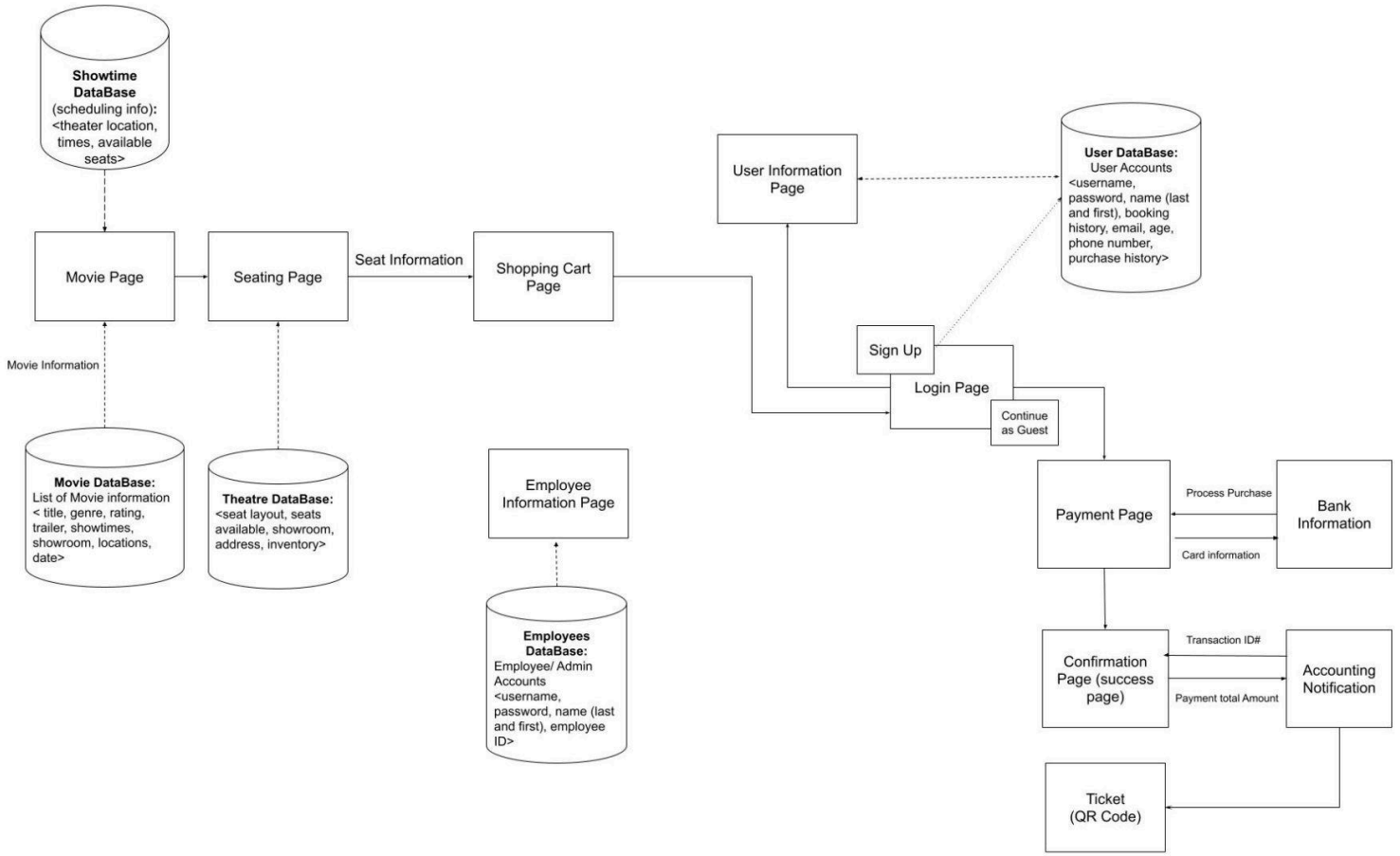


Diagram Description:

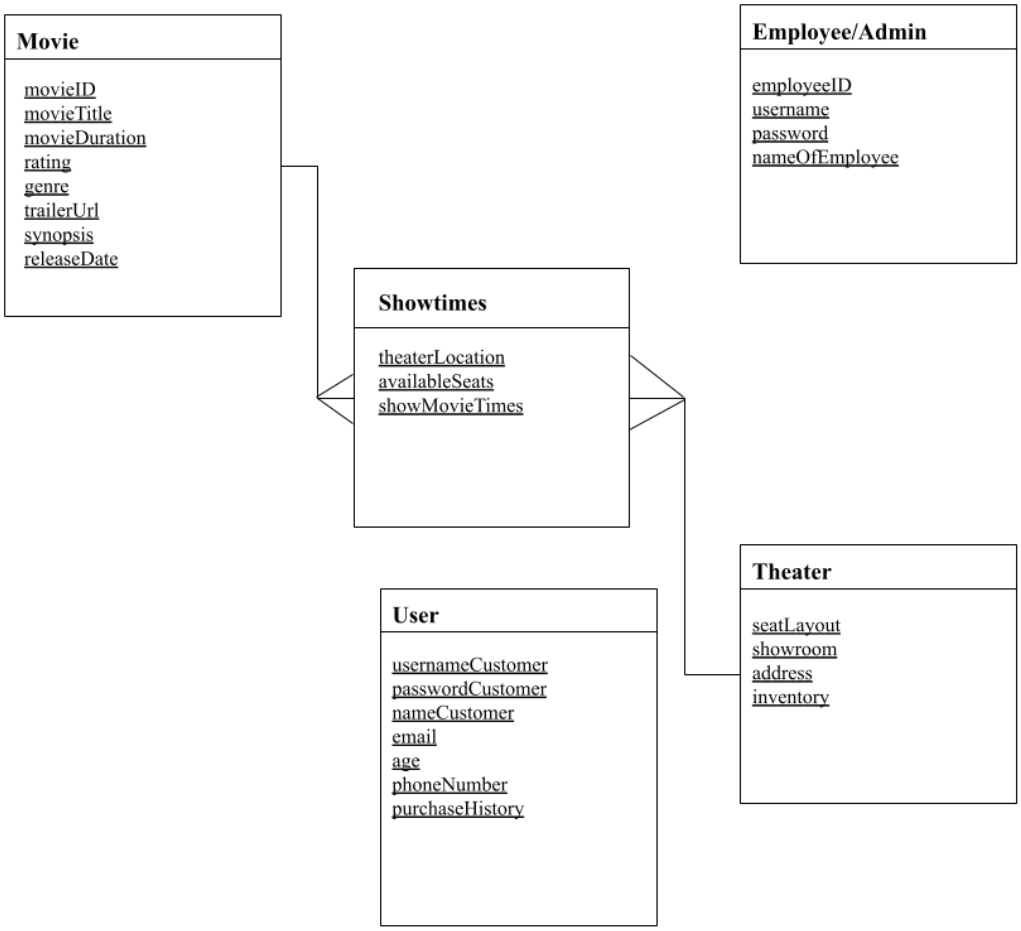
The diagram begins when users access the movie page accessing the movie database. When users select a movie the system will query the movie database to receive details about that move. Then users are prompted to select a showtime, this is when the system will query the showtime database and check for available showtimes and check if the corresponding seats are available. Once a user selects desired items they are led to the shopping cart where they are able to confirm their purchase and continue onto the login/register/ guest page where users will be given the option to login to an existing account, register for an account or continue on as a guest. The existing users information is stored in a user database and if the user chooses to login with an already existing account then the system will query the user database for said information. Next users will reach the payment page where they will enter their payment information, since we will not be storing their payment details there is no need to link the user database for this portion. As soon as the payment transaction is successful the user is sent to a confirmation page with the ticket and details of the movie tickets they purchased. When this confirmation screen is reached a message to accounting is made notifying them of a purchase. For the employee and admin database, these are responsible for storing information about employees and administration that manage the ticketing system and theater operations.

6.2 Data Management Strategy

6.2.1 Data Dictionary and Database Relationships

Data Dictionary

movieID: Unique identifier assigned to each movie
movieTitle: The title of the movie being shown
movieDuration: The total length of the movie in minutes
Rating: The films ratings based on customer reviews
Genre: The category or type of movie based on a theme
trailerUrl: The URL of the movie trailer
Synopsis: A quick summary of the movies plot
releaseDate: The official release date of the movie
theaterLocation: The location or address of the movie theater
availableSeats: The number of unreserved seats that are available for a movie showing
movieTimes: A lost of available times to watch a specific movie
employeeID: A unique identifier assigned to each employee
Username: The employees or system admins username
Password: The employee or admins password associated with username
nameOfEmployee: The full name of the employee
usernameCustomer: Customers unique username for logging into the system
passwordCustomer: What is needed in order to access customer account, associated with customer username
nameCustomer: The full name of the customer
Email: The customers email
Age: The customer's age
phoneNumber: The customer's phone number
purchaseHistory: A record of past movie purchases made by customer
seatLayout : The arrangement of seats at a certain location
Showroom: Specific theater in the movie theater where movies are shown
Address: The physical address of the theater
Inventory: A record of available tickets, food or services in the movie theater



The data dictionary describes the different instances in the database. The diagram above shows the relationship between each database. The movie database is the most important because most databases cannot work properly without it.

Movies and showtimes have many-to-many relationships because each movie can have many showtimes.In the same way, each showtime can have many movies playing depending on the theater location. The showtimes database has a many-to-one relationship with the theater database. The reason for this is there are different showtimes for each theater location. The employee and the user database work independently on the system.

6.2.2 SQL Diagram

Movies

id	title	duration	releaseDate	rating	genre	trailerUrl
12785	Venom: The Last Dance	1 hr 49 min	2024	4 stars	Action, Sci-Fi	https://blossomtheaters.com/yhscw66f
46574	Smile 2	2 hr 7 min	2024	4 stars	Horror	https://blossomtheaters.com/xyasbw23f
89765	Look Back	1 hr 15 min	2024	5 stars	Animated, Drama	https://blossomtheaters.com/abzge987f

Theaters

id	theater	numberOfShowrooms	seatQuantity	address	screenTypes
12356	Blossom Theaters SD	10	50	1982 Westview Pkwy, San Diego, CA 92126	IMAX 2D, IMAX 3D, 4DX
90873	Blossom Theaters Chula Vista	8	50	5678 Tierra Del Ray, Chula Vista, CA 91910	IMAX 2D, IMAX 3D
56783	Blossom Theaters Poway	8	50	1234 Poway Rd, Poway, CA 92064	IMAX 2D, IMAX 3D

Showtimes

showMovieTimesID	movieID	theaterID	seatsID
7PQ7R	12785	12356	54557
56TVW	46574	90873	64453
34RTAI	89765	56783	89721

Employees

EmployeesID	Username	Name	Actions
EMP001	j.doe12	John DOe	[Edit][Delete]
EMP002	jsmith	Jade Smith	[Edit][Delete]
EMP003	k.mar	Keith Mar	[Edit][Delete]
EMP004	ray.hart	Rey Hart	[Edit][Delete]

User Database

userID	username	Password (Encrypted)	fullName	emailAddress	age	phoneNumber	bookingHistory (Movie IDs)
13APD	BlossomShine	********	Nicole Smith	nicolesmith@gmail.com	49	(619)548-8494	12785
K93M5	SmileyFace	**********	Jane Doe	janedoe@Yahoo.com	26	(858)629-4853	46574
9O1ED	CosmicKnight	**********	Malik Rivera	malivrivera@gmail.com	18	(619)803-4092	12785

The project has opted to adopt SQL (Structured Query Language) due to its superiority in maintaining relationships across datasets, guaranteeing data integrity through primary and foreign keys, and streamlining data operations such as searching, changing, and retrieving records. For managing interrelated elements like movies, theaters, showtimes, employees, and users, SQL databases like MySQL or PostgreSQL are perfect since they are extremely scalable,

standardized, and extensively maintained. A single centralized database was selected to streamline data management, reduce redundancy, enhance performance, and simplify backup and recovery processes. Based on functionality, the data is divided into the following tables: Movie (for storing information such as title and genre), Theater (for managing location-specific data), Showtimes (for organizing movie and theater schedules), Employee (for managing staff roles and assignments), and User (for tracking customer interactions). By keeping critical information, such as passwords, in unique and secure tables, this modular design guarantees readability, effective query performance, scalability for future improvements, and robust security.

6.3 Data Management Constraints

The user's information and the employee information are separate because it makes the data safer. The employee would need to have consent or authorization from the user to access any information that they have. The user's information and data is more prone to be leaked because the user could share their information with elsewhere or whoever they choose. However, the employee's information would be confidential, so the employee's information would and should not be shared with anyone else, even coworkers. This means that the employee's information would be unable to be accessed unless by the employee themselves or the higher ups who would have the ability to access the employee's information. The movie insertion would be used the same way every time by the employees so that it is easy to manipulate and hard to be accessed due the the SQL in which it makes it easier to store the data that is structured where all the movie information would be stored, entered along with formatted.