Group 4's "Blossom Theaters" SRS document is thorough and demonstrates a solid understanding of the system's requirements. The functional and non-functional requirements are well-structured and cover all necessary aspects of the ticketing system. However, the inclusion of visual aids such as use case, sequence, or data flow diagrams would significantly enhance the document by providing a clearer understanding of the system's processes and interactions. Overall, it's a strong submission that could be further improved with the addition of diagrams and slightly more abstraction in the introduction. Great job!

Zoom Link:
https://us05web.zoom.us/j/3178519748?pwd=bpbrlaP5dDRX97QiHQOsAVrXAzECd1.1

URL of GitHub: Intro to Software Systems-Rita Yousif
https://github.com/ryousif1694/Group-4?fbclid=IwY2xjawFv4gxleHRuA2FlbQIxMAABHTR_sKOjDHLmj3UOCH8mEw1hbtIvfjHe_6WAZWk6l1bG9P0fQT3Q5x7eBA_aem_vEGRvfBD3P2ZoPpxvgjwyA

Jesyl Zendejas
Judge Dred Banal
Abigail

ryousif1694@sdsu.edu
gfhanna@sdsu.edu
pgogawale0410@sdsu.edu

Tasks: Note whatever you do from here, You have to do to both UML and SWA along with a description of what you have.

Payment: Jesyl
Registering: Register account for a guest or customer (Page):Jesyl
Location of Multiple showrooms : show how many seats in each of the showrooms (50 seats): Abigail
Admin: Judge
Employee Info: Rita
User Info: Rita

<Blossom Theaters>

Ticket: seat number show time date movie title any purchases and customer name and ticket ID : Rita
Shopping Cart: Judge
Movie: rating, genre, title, search, filter: Abigail
Design constraints: Judge

Architectural diagram of all major components: Jesyl and Abigail

UML Class Diagrams: Name

Description of classes:

Description of attributes:

Description of operations:

Descriptions should be detailed and specify datatypes, function interfaces, parameters, etc..:

Blossom Theaters

Software Requirements Specification

Version 1.0

9/23/2024

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	Author	Comments
<date>	<Version 1>	<Your Name>	<Developed Requirements >
	Version 2		Added design to the document

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 Specificati 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.

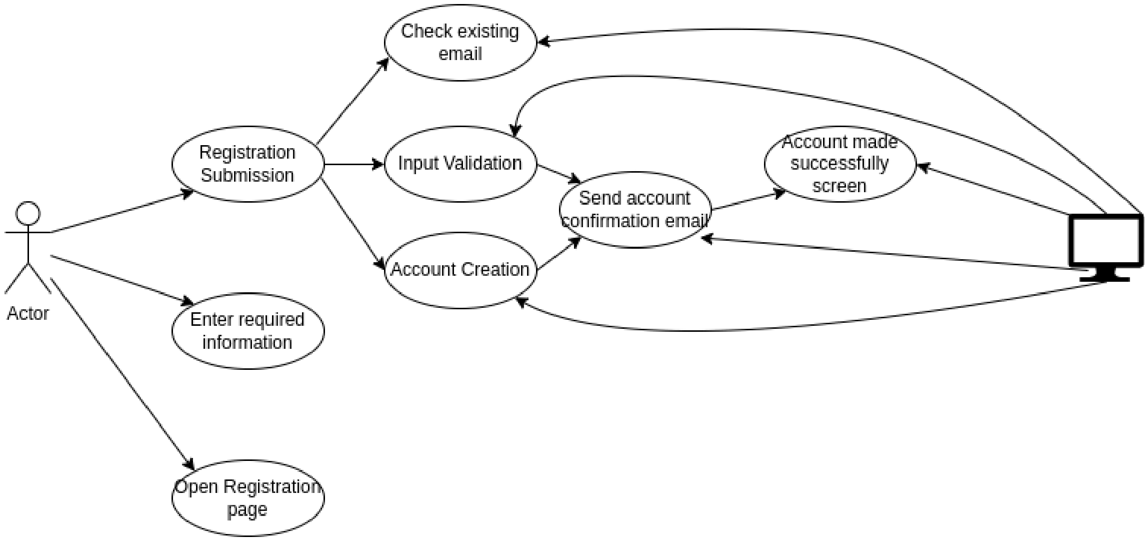
Actor: User, system

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

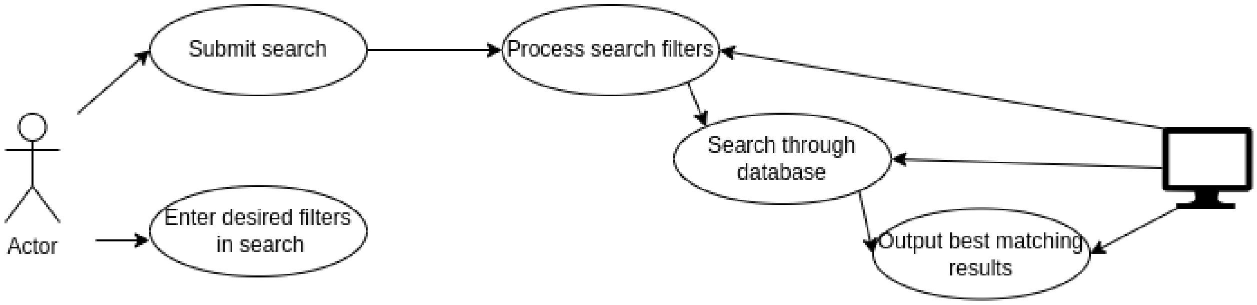
Conditions:

- **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

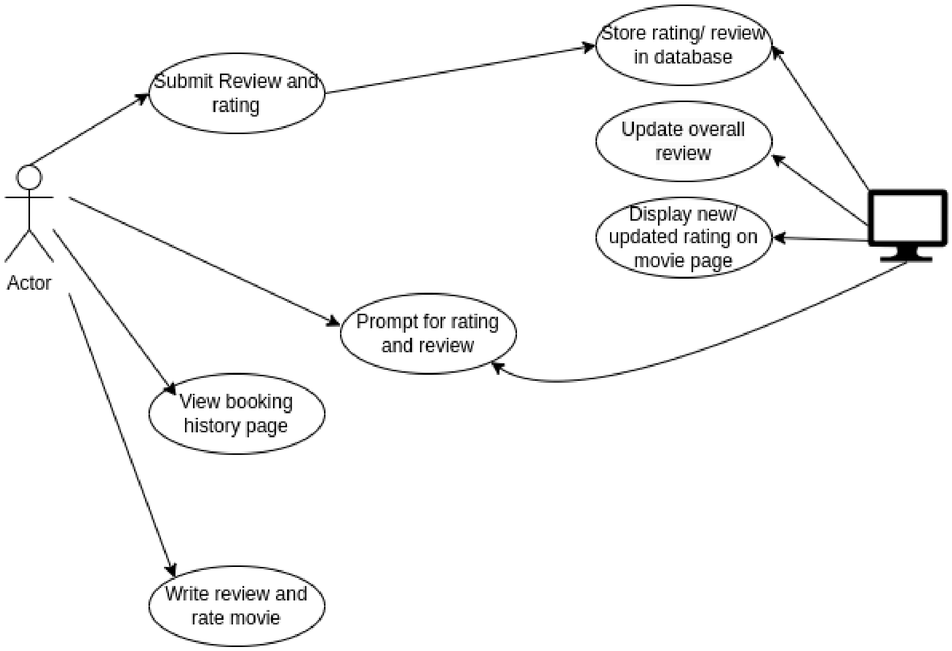
Actor: User, system

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


Submit Review and rating
Store rating/ review in database
Update overall review
Display new/ updated rating on movie page
Actor
Prompt for rating and review
View booking history page
Write review and rate movie

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

Section 4

Movie Class

Class Description:
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. This class will also serve as the blueprint for the admins when they enter information about the movie they will show in the theater.

Attributes Description:
movieTitle - stores a string value that represents the name of the movie
movieGenre - stores a string value that represents the genre of the movie. The genre can be horror, action, comedy, documentary, and many more.
releaseDate - stores an integer value that represents when the year the movie was released
rating - stores an integer value that represents the rating of the movie out of a five star rating system
cast - stores a list of string values representing the cast members of the movie
synopsis - stores a string value that represents a short summary of the movie
trailerUrl - stores a string value that represents the url link to view the trailer of the movie
showDate - stores an integer value that represents the dates the movie will be showing on Blossom Theaters
showTime - stores an integer value that represents the times the movie will be showing on Blossom Theaters
location - stores a string value that represents which Blossom Theater location the movie will be playing
Showroom - stores an integer value that represents the movie room number the movie will be playing based on the selected location
price - stores an integer value that represents the price of each movie ticket for the movie
seats - an integer value that tells the number of seats available for the movie based on the selected location and showRoom

Operations Description:

getMovieTitle(): String

- The function returns a string that represents the title of the movie

setMovieTitle(String): void

- The function takes a parameter of data type String, which will be used to assign or update the title of the movie

getMovieDetails(): String
setMovieDetails(string, int, int, List<string>, string, string)
addCast(String): void
removeCast(String): void
getViewingTimes(): String
setViewingTimes(int, int, string, int): void
getPrice(): int
setPrice(int): void
addAvailableSeats(int): void
removeAvailableSeats(int): void

Location

Attributes:
name: String
address: String
availableMovies: List<Movie>
showRoom: int
capacity: int
availableSeats: int

Operations:
updateAvailableMovies(List<Movie>): void
updateShowRoom(string, string): void
updateCapacity(string, int): void
availableSeatsCounter(int): int
isSoldOut(): boolean

Payment:

transactionID: Integer

totalAmount: Float

paymentMethod: String

paymentProcessingStatus: String

timeDate: ctime

userTicker: Ticket

seatID: String

processPayment(totalAmount: Float, paymentMethod: String, updateSeat (ticket): Void): Boolean

refundingPayment(transactionID: Integer): Boolean

userPaymentDetails(transactionID: Integer): String

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

updateSeat (ticket): Void

OverView of Class:

Description of 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

Descriptions of 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).

Registering: Register account for a guest or customer (Page):

Registration (class):

Username: string

Password: string

Email: string

firstName: string

lastName: string

Age: integer

processingStatus: string

userType: String

phoneNumber: integer

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

validateEmail(email): Boolean

validatePassword(password): boolean

getRegistrationStatus(): string

emailConfirmation(): Void

seeUserType(): String

OverView of Class:

Description of 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.

Descriptions of 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.

SWA Diagram:

Login:

If a returning user is logging in, it will check if the username and the password match what is in the database. Depending on whether or not they fulfill the criteria the user will either be met with the continuing payment page or the page will rest, prompting them to re enter their username and password, telling them something was input incorrectly.

Sign-Up:

This gives the option to sign up for the account and this information will be stored in the database

Continue as Guest:

This will allow users to just continue on without having to fill anything other than an email and name (for the purpose of sending a ticket).

Purchase:

This is where the user will input their form of payment and start to fill in their card information. This would consist of the information like; card number, CVV number, expiration date of card, name as it appears on card. This page is necessary for every user to enter, no matter if it is a guest, new user or returning user. Since we will not be storing user payment information for security purposes, they will not be stored in the database.

Once the information is entered the process is started with checking if the card information is correct then sending off a payment request to the bank. If this request is processed and successful then it will lead to the next page: Confirmation page. If the payment fails then the page will reload and ask the user to re enter their payment information.

Confirmation page: this will consist of a success message as well as a message pertaining to a copy of the ticker being sent to their email.

Accounting notification: this will notify the accounting that we made a sale, this will also allow them to see what was bought and take inventory into account

Ticket Class: 2-3 sentences explaining attributes and operations

Ticket Attircutes:

seat:(String)-The customer will be able to view there seats in which it is a parameter that involves the seating row and column.
showtime:(int, int)-
movieTitle: String-
price: double-
foodsAndBeverages:String-
ticketID: int-
ticketQRCode:String-

Operation:

User info

updateProfileInformation:(int,int):int//replace old int with new int which updates it to the int so returns int
chooseMovie(Movie):void//just does the action without any updates or returns
addMovieToShoppingCart(ShoppingCart):void
watchMovieTrailers():void
cancelTicket(Ticket):void