

# ABC Theater's Movie Ticketing System

## Software Requirements Specification

Version 1

September 26, 2024

Group #6

Cameron Lee, Roger Dao, Ross Hackett,  
Matthew Dominguez

Prepared for

CS 250- Introduction to Software Systems

Instructor: Gus Hanna, Ph.D.

Fall 2024

## Revision History

Date	Description	Author	Comments
09/26/24	Version 1	Cameron Lee, Roger Dao, Ross Hackett, Matthew Dominguez	Added introduction, general description, specific requirements (excluding design constraints, logical database requirements, and elaboration on other requirements)

## Document Approval

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

Signature	Printed Name	Title	Date

# Table of Contents

<b>Revision History.....</b>	<b>3</b>
<b>Document Approval.....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Definitions, Acronyms, and Abbreviations.....	1
1.4 Overview.....	1
<b>2. General Description.....</b>	<b>1</b>
2.1 Product Perspective.....	2
2.2 Product Functions.....	2
2.3 User Characteristics.....	3
2.4 General Constraints.....	3
2.5 Assumptions and Dependencies.....	3
<b>3. Specific Requirements.....</b>	<b>4</b>
3.1 External Interface Requirements.....	5
3.1.1 User Interfaces.....	5
3.1.2 Hardware Interfaces.....	5
3.1.4 Communications Interfaces.....	5
3.2 Functional Requirements.....	6
3.2.1 Ticket Purchasing.....	6
3.3 Use Cases.....	7
3.3.1 Use Case #1: Customer Movie Search.....	7
3.3.2 Use Case #2: Customer Login.....	7
3.3.3 Use Case #3: Customer Ticket Purchasing.....	8
3.3.5 Use Case #4: Customer Support Contact.....	9
3.4 Classes / Objects.....	10
3.4.1 Class / Object #1: Customer.....	10
3.4.2 Class / Object #2: Theater.....	10
3.4.3 Class / Object #3: Ticket.....	10
3.5.1 Performance.....	11
3.5.2 Reliability.....	11
3.5.3 Availability.....	11
3.5.4 Security.....	11
3.5.5 Maintainability.....	11
3.5.6 Portability.....	12
3.6 Inverse Requirements.....	12
3.7 Design Constraints.....	12
3.8 Logical Database Requirements.....	12
3.9 Other Requirements.....	12
<b>4. Analysis Models.....</b>	<b>12</b>
4.1 Sequence Diagrams.....	12
4.3 Data Flow Diagrams (DFD).....	12

4.2 State-Transition Diagrams (STD).....	12
<b>5. Change Management Process.....</b>	<b>12</b>
<b>A. Appendices.....</b>	<b>13</b>
A.1 Appendix 1.....	13
A.2 Appendix 2.....	13

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to provide adequate understanding to software engineers and web designers of the functionality and features needed to be implemented for the product as well as a description of the intended user experience the product should deliver, such as navigation of the website, logging/signing in, payment and means of receiving tickets.

## **1.2 Scope**

The product will be an online movie ticketing system that allows users to browse films showing at participating locations. Users should be able to search by three methods

1. Browsing films sorted by popularity within a user selected location's selected radius.
2. Searching for specific films and all available showings within a user's selected radius.
3. Browse all films showing at a user selected theater.

Users should be able to purchase tickets via Paypal, Venmo, Affirm, and Shoppe, and receive tickets and payment confirmation via email used to login.

## **1.3 Definitions, Acronyms, and Abbreviations**

There are no relevant keywords or uses of language contained in this document that should require defining.

## **1.4 Overview**

The rest of the SRS contains general information on how the product should be organized and function followed by more specific details on how tasks the application should be able to perform and handle are meant to be structured and implemented. The SRS is organized as so to provide a general idea of how the finalized product is intended to look before delving into specific outlines for its individual characteristics.

# **2. General Description**

Online ticket purchasing software (website) for ABC theater's 6 physical locations (they all exist within San Diego.)

The product will allow customers to conveniently remotely reserve seats and purchase tickets online from one of six ABC Movie locations.

Search function with ability to search for ABC Theater by location (zip code), search for a theater with showings of a particular movie by specifying the movie, search by time to view showings at theaters at that certain time, general movie availability showtimes and more..

Our website is designed keeping in mind the things needed to effectively purchase tickets in a rush or in advance.

Customers can register on ABC theater's website to save preferred location, payment methods, view purchased tickets easily within the same site, expediting the theater booking process. Tickets can be purchased as a guest or a user, customers will receive via email a receipt for their order that includes their digital ticket that can be printed or serve as an entry ticket using a mobile device once scanned via QR code.

Tickets are unique to prevent fraud.

Payment methods include most accepted credit/debit cards, and other popular payment methods including Paypal, Venmo, Affirm, and Shoppe.

Customers can submit tickets to receive customer support for issues related to the website's functionality, user account credentials, order details, and order cancellation.

Customer support representatives can login using their credentials and review submitted tickets, cancel orders, issue refunds, or make adjustments to customer's purchases and/or registered account details.

## **2.1 Product Perspective**

Independent and self-contained ticket purchasing website designed to be accessible on common mobile and desktop devices. The main goal is to provide convenience for patrons wanting to order tickets and reserve seating in advance, or patrons looking to quickly and easily order tickets. After purchase, printable tickets including scannable QR codes will be emailed to the customer promptly.

Considerations are made to handle increased traffic surrounding popular movie releases, and ticket limits are enforced to limit the act of scalping tickets. For ease of use for both patrons and staff in case of an outage, popular and easily verifiable digital payment methods will be accepted as well as commonly used credit/debit cards, the same as in our theaters.

The ticket purchasing system is meant to assist in the daily operations of ABC Theater's business and is designed to be cohesive with preexisting in person operations. Ease of use and identical function to kiosks and booths is a priority to cut down on additional training and limit burden on staff and maintenance personnel at theater locations in the case of system failure. The simplicity of the design is to avoid limiting operations in case of outages, unique threats to the system, including online threats, server downtime, etc. (single point of failure)

## **2.2 Product Functions**

Summary of the functions the software will perform:

To register as a user

Login using credentials of an account registered with ABC movies.

Search for movies by title. Will return locations, showtimes, etc.

Search of theaters by location (zip code)

Return list of movies playing at a theater location

Return list of available showtimes for a theater location

Reserve a ticket for purchase for 10 minutes.

Purchase tickets.

Purchase tickets as guest

Purchase tickets as registered user

Receive customer support for customer account assistance, payment and ticket assistance

Customer Support Representative access to: customer account information, management of payment processes (in the case that refunds or adjustments need to be made.)

Send ticket to email (to resend if there is a need to, will provide the purchased ticket(s) as scannable digital QR code and/or printable ticket)

## **2.3 User Characteristics**

General characteristics of the eventual users of the product that will affect the specific requirements:

Users of the product: People with the desire and ability to purchase tickets online to skip the line in person, reserve tickets and seating ahead of time,

Education level: In or after high school (they are debit/credit card holders)

Experience: By this point they will have some or no experience with ordering online. Our site is simple to use and we maintain in person and over the phone sales taking this into account.

Technical expertise: By this point the user will have sufficient experience with using their personal computer or phone for similar purposes, if not, our site is simple to use and we maintain in person and over the phone sales taking this into account. (such as the customers who do not go online much, users of devices we don't maintain support for)

## **2.4 General Constraints**

Reliability requirements: we have designed our system to be redundant with the functions of in person operations.

- We don't accept cryptocurrency.

- We send confirmation of purchase including the printable ticket & QR code to emails only, customers ordering via phone will have to pick up tickets in person with verification.
- We only list prices in USD as part of our design (our locations are in San Diego only)

Safety and security considerations:

- You need to use a browser, we don't have an app. There is too much overhead related to arbitrary phone operating systems updates involved in maintenance of an app. We choose to use browsers because of this.
- Adherence to protocol involving storage of personally identifying data, including information used in user account registration, payment methods, etc.

As a business, we want to maintain stability of our operations and maintaining expanded functionality of our website contradicts that.

## **2.5 Assumptions and Dependencies**

Assumptions:

Designed to be operated through a desktop/laptop or phone's browser. As such, as long as the device is suitable for day to day use, it is (powerful enough to run youtube on a browser/app, access emails)

The user will encounter frustration if they are accustomed to ordering over the phone and receiving their purchased item in person

The employees will receive the necessary training on how to operate

Dependencies:

We are designing for personal devices x86 desktop/laptops and android/iOS. The user will be using Android, iOS, Windows, macOS, Linux on their personal device.

Infrastructure at our theater locations will require maintenance and support to support

Including training of employees to handle increased functionality, and instructions on what to do if online functions cannot be reached.

## **3. Specific Requirements**

### **3.1 External Interface Requirements**

#### **3.1.1 User Interfaces**

Homepage will have a clear display of available movie options, as well as a search function.

Movie thumbnails can be clicked on to access more information including summary, showings, and location. Thumbnails can also be scrolled horizontally for browsing.

Search results can be filtered by rating and genre

Seat map will display when selecting seats and purchasing tickets.



### **3.1.2 Hardware Interfaces**

Mobile version of the homepage can be interacted with touch. Scrolling feature is interactive with physical scrolling.

Movie times are accurate to the location and time of the Customer when accessing from Phone or desktop.

The website will be simple and require low data transfer from devices connected to the internet for fast, effortless, and easy use.

QR codes will be generated and can be physically scanned through phone screens at the appropriate cinema.

### **3.1.3 Software Interfaces**

Connectivity to send SMS or Emails to customers for Receipts, tickets, or promotions.

Live update of tickets and seats available.

Up to date showtimes of movies of up to two weeks, and refreshes every week. Past times are removed from availability.

Third party payment integration with Paypal, Venmo, affirm, and Shoppe.

### **3.1.4 Communications Interfaces**

Communication to API for showtimes.

Having stable security to handle private information between server and user.

Integration with the server to send SMS and emails immediately.

## **3.2 Functional Requirements**

This section will outline the key features for

### **3.2.1 Ticket Purchasing**

#### **3.2.1.1 Introduction**

This feature allows Customers to purchase tickets and seating from a selected showtime.

#### **3.2.1.2 Inputs**

- Select Movie
- Select Time
- Select Seating
- Choose payment option

#### **3.2.1.3 Processing**

1. The system retrieves available tickets and seating information
2. Customer selects time and seat
3. The system will retain this information temporarily
4. Customer completes purchasing prompts
5. Ticket is reserved with their information, and is removed from list of availability

#### **3.2.1.4 Outputs**

- Movie information
  - Showtimes
  - Location
  - Price
  - Summary
  - Available seats
- Ticket with Barcode/QR code
  - Delivered through Email and SMS
- Receipt for proof of purchase
  - Delivered through Email and SMS, and temporarily stored on server

#### **3.2.1.5 Error Handling**

1. If the Customer is inactive for 10 minutes, the temporary seat data is removed and the Customer must start again.
2. If the chosen seat is unavailable after purchase, the Customer is redirected to choose new seats.
3. If payment is unsuccessful, the Customer is redirected the cart to retry with another payment method

### 3.3 Use Cases

#### 3.3.1 Use Case #1: Customer Movie Search

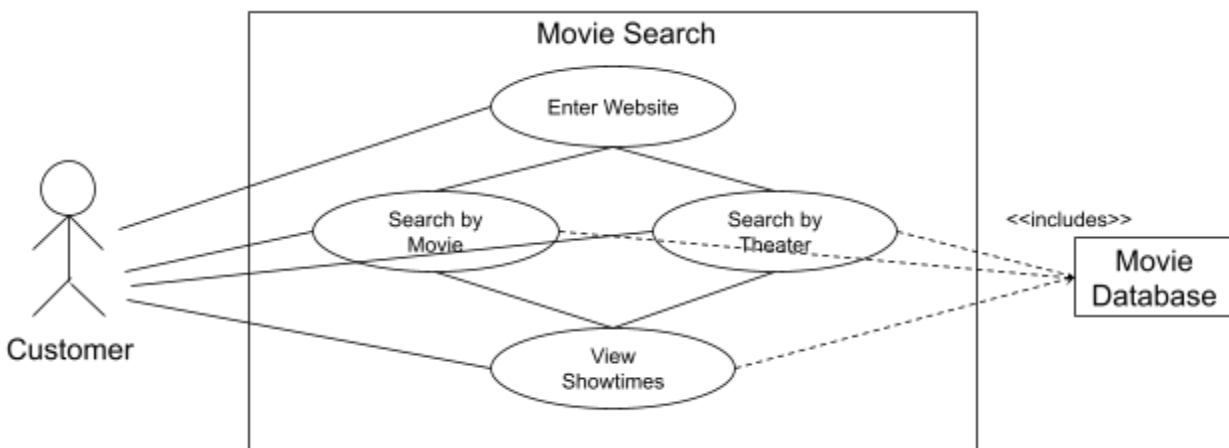
**Actors:** Customer

**Flow of Events:**

1. Customer accesses movie theater website from a browser.
2. Customer is greeted with a search bar along with current newly released movies.
3. If Customer searches by movie, the website will display a list of ABC Theater locations and their subsequent showtimes for the specified movie. Showtimes are categorized by Standard, Premium, and 3D theaters.
4. If Customer searches by theater then they will see a similar interface but with showtimes of all movies available at the selected location.
5. If no showtimes are available for the selected movie or theater the customer will be prompted with a message detailing so.
6. Clicking on a showtime will redirect the customer to the ticket purchasing case.

**Entry Conditions:**

1. Computing requirements: supported browser.



#### 3.3.2 Use Case #2: Customer Login

**Actors:** Customer

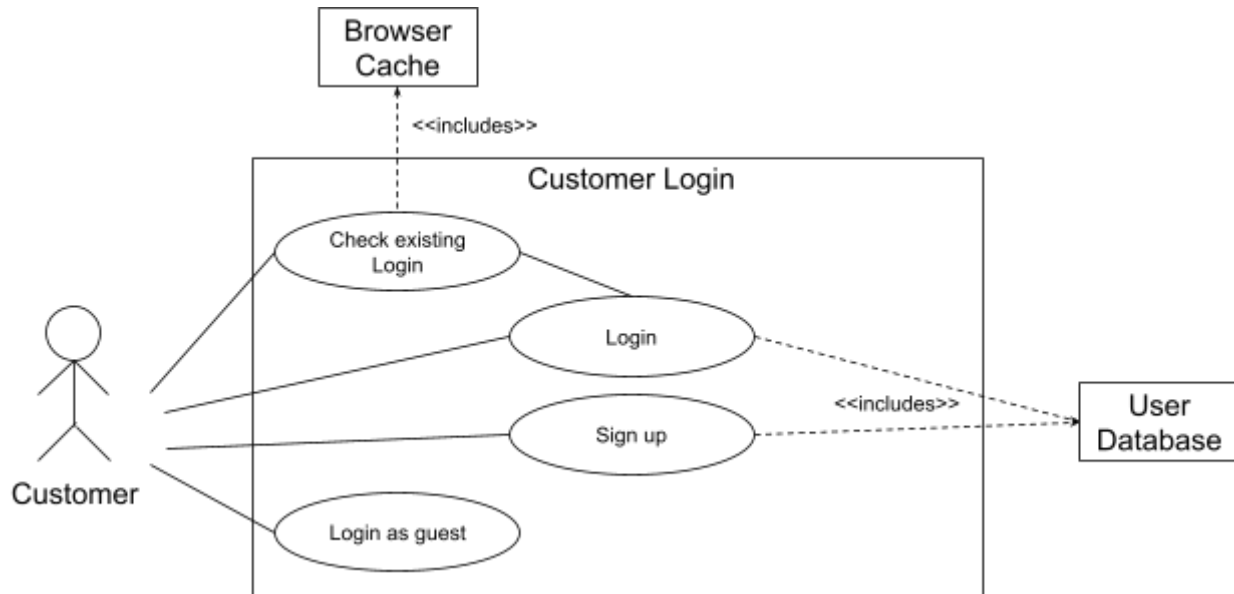
**Flow of Events:**

1. Customer attempts to purchase a ticket.
2. The website checks the browser cache for a stored login. If a login is found, the next events are skipped and the customer proceeds to ticket purchasing.
3. If no existing login is found, Customer is prompted with 3 options,
4. Customer selects either to login, create a new account, or continue as a guest.
  - 4.1. If continue as guest is selected, proceed with ticket purchasing.
  - 4.2. If login is selected, Customer is prompted to enter a username and password.
  - 4.3. If "create new" account is selected, Customer is prompted to enter a username, password, valid email, and valid phone number.

5. Website server queries database to validate the login or to check if an account already exists using the same username or email.
6. If login credentials are invalid, Customer is prompted to try again or reset their password.
7. Customer continues to purchase tickets.

**Entry Conditions:**

1. Customer attempts to purchase ticket.



### 3.3.3 Use Case #3: Customer Ticket Purchasing

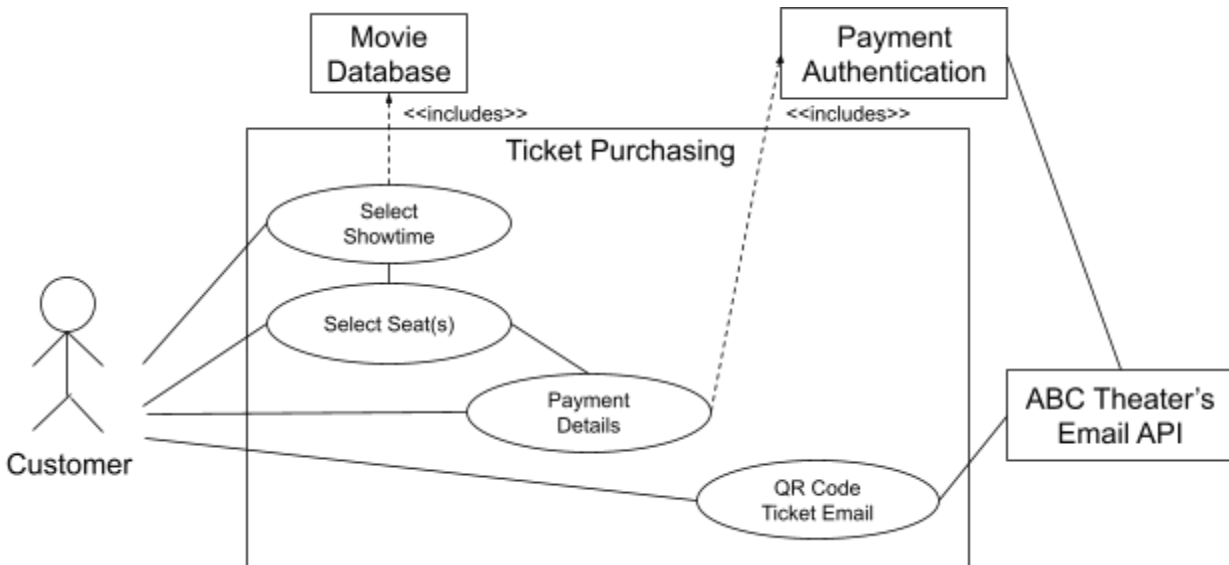
**Actors:** Customer, Third-Party Payment Processor (Apple, Google, Banks, etc)

**Flow of Events:**

1. After selecting showtime and logging in, the website accesses the database to see what seats are available for the selected showtime.
2. The customer can then select what seats they would like to purchase from a visual interface showing a top down view of the theater.
3. As the seats are selected, the tickets are added to the customer's cart. There is no limit to how many tickets from the same theater a customer can select.
4. Once in the cart, the website tells the database to hold these tickets. A 10 minute countdown begins on the webpage. If the user does not checkout before the 10 minute timer ends, these tickets are made available to other customers, and the Customer is notified that their seat reservations have expired.
5. Selecting checkout redirects customers to a payment screen. Forms of payment accepted are Apple pay, Google pay, and all major credit cards.
6. Payment information is verified with a third-party payment processor. The payment processor sends a confirmation or denial of payment based on the card's status.
7. Once verified, the customer will receive an email from ABC Theater's with their tickets and QR codes for entry.

**Entry Conditions:**

1. Customer has selected a theater, movie, and showtime.
2. Customer has valid payment method to purchase tickets.



### 3.3.5 Use Case #4: Customer Support Contact

**Actors:** Customer, Customer Support Representative

**Flow of Events:**

1. A Customer may send a message to the ABC Theater's customer support team if they have any issues regarding the use of the website, a previous purchase, or any other technical problem.
2. From the home page of the website, Customer can scroll down to the bottom and select help
3. Customer will be shown a series of frequently asked questions and their answers.
4. If these don't help, a button for "Submit a Ticket" is available.
5. Once selected, Customer will be prompted to submit their name, phone number, email address, and write a message detailing their issue. An option to attach images is also available, with a file size limit of 350mb.
6. A Customer Support Representative will review the submission, and respond within 24-48 hours via phone call or email.
7. The Customer Support Representative is able to access customer account information as well as manage payment processes in the case that refunds or adjustments need to be made.

**Entry Conditions:**

1. Customer needs help with services provided by ABC Theaters.
2. Customer Support Representative is available to respond to tickets.

## 3.4 Classes / Objects

### 3.4.1 Class / Object #1: Customer

This class represents a customer interacting with the website

3.4.1.1 Attributes:

- **customerID:** An identification number used in the database for each customer.

- **username**: The customer's chosen username.
- **password**: Encrypted password for the user login.
- **email**: The customer's email address for contact and promotions.
- **phoneNumber**: The customer's phone number for notifications and promotions.

#### 3.4.1.2 Functions:

- **searchMovie(movieTitle)**: Allows the customer to search for movies by title.
- **searchTheater(zipCode)**: Allows the customer to find theaters near them based on zip code.
- **login(username, password)**: Authenticates the customer based on credentials stored in the ABC Theater's user database.
- **holdSeat(showtimeID, seatNum)**: Reserve's seat for the user and begins 10 minute countdown.
- **purchaseTicket(showtimeID, seatNum)**: Initiates

#### References to functional requirements:

- Use Case 3.3.1: Customer Movie Search - utilizes searchMovie() and searchTheater().
- Use Case 3.3.2: Customer Login - needs all variables and login() function.
- Use Case 3.3.3: Customer Ticket Purchasing - uses holdSeat() and purchaseTicket().

### 3.4.2 Class / Object #2: Theater

#### 3.4.2.1 Attributes:

- **theaterID**: Identification number used in database for each ABC Theater location
- **theaterName**: Name of the location used for searches.
- **address**: Street address of the theater.
- **showtimes**: A list of movies and showtimes playing at this theater.

#### 3.4.2.2 Functions:

- **getShowtimes()**: Returns the available showtimes for this theater.
- **getMovies()**: Returns the list of movies playing at the theater.

#### References to functional requirements:

- Use Case 3.3.1: Customer Movie Search

### 3.4.3 Class / Object #3: Ticket

#### 3.4.3.1 Attributes:

- **ticketID**: A unique code for each ticket.
- **movieTitle**: The title of the movie the ticket is for.
- **showtimeID**: The specific showtime for the movie.
- **seatNum**: The seat assigned to the ticket.
- **price**: Price of the ticket.
- **isPurchased**: A boolean for tracking if the ticket has been purchased or not.

#### 3.4.3.2 Functions:

- **reserveTicket(showtimeID, seatNumber)**: Reserves the ticket for purchase.
- **purchaseTicket(paymentInfo)**: Processes the ticket purchase.
- **sendTicket(email)**: Sends ticket QR code to the ticket recipient.

#### References to functional requirements:

- Use Case 3.3.3: Customer Ticket Purchasing

### 3.5 Non-Functional Requirements

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

#### 3.5.1 Performance

- Response Time: Search results for movies and theaters should load within 3 seconds.
- Ticket Purchase Time: Payment confirmation and receiving the ticket via email should take no longer than 20 seconds after the customer submits their payment.
- Concurrent Users: The website should be able to handle up to 2,000 concurrent users browsing and searching for movies at the same time without experiencing performance impacts.
- Seat Availability Refreshes: Seat Availability should refresh constantly and be updated in real time to ensure that the same seat is not purchased by two different people.

#### 3.5.2 Reliability

- Uptime: The system should guarantee a 99% uptime from 6am to midnight.

#### 3.5.3 Availability

- 24/7 Access: The system should be available 24 hours 7 days a week, including holidays.
- Peak Traffic: The system should remain fully functional when experiencing high traffic, such as opening nights of popular movies.

#### 3.5.4 Security

- Data Encryption: All sensitive customer data (login info, credit card details, and personal information) should be encrypted using industry-standard protocols (i.e. AES-256).
- Payment Security: Payment processing should comply with Payment Card Industry Data Security Standard to protect customers' financial data.

#### 3.5.5 Maintainability

- Modular Code: The system should be designed with a modular codebase to ensure the individual components can be updated without impacting the rest of the system.
- Version Control: Use of version control during development should be utilized to ensure backups and collaboration in case of unexpected errors.
- Documentation: Documentation for all system components should be maintained and updated regularly by the developers.

#### 3.5.6 Portability

- Cross-platform: The system should be accessible from various different desktop browsers (Chrome, Firefox, Safari, Edge) as well as mobile browsers on Android and iOS.

### 3.6 Inverse Requirements

- No Unauthorized Access: The system must not allow any customer or third party to access databases without proper authorization.
- No Slowdowns During High Traffic: The system must not experience impacted performance during peak usage.
- No Double Ticket Bookings: The system should not allow for two customers to book the same seat for the same showtime.
- No Unencrypted Data Storage: The system must not store sensitive customer info in any unencrypted formats.
- No Excessive Downtime: The system must not be unavailable for more than 1% of the time.
- No Exposure of Sensitive Information: The system must not display any sensitive customer information in plain view.

### 3.7 Design Constraints

*Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

### 3.8 Logical Database Requirements

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

### 3.9 Other Requirements

*Catchall section for any additional requirements.*

## 4. Analysis Models

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.*

### 4.1 UML Diagrams

diagram be here

Class Name:

Purpose:

Attributes:

Methods:

Description:

### 4.2 Software Architecture Diagram (SWA)

Diagrams and descriptions here



## Development Plan and Timeline:

Cameron Lee

- section 3 part bottom

Roger Dao

- <https://github.com/rogerdao>
- section 2

Ross Hackett

- section 1

Matthew Dominguez

- section 3 part top

## 4.3 Data Flow Diagrams (DFD)

## 5. Change Management Process

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

### A. Appendices

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

#### A.1 Appendix 1

#### A.2 Appendix 2