

# **Software Requirements Specification**

for

## **Project-Theater**

Version 1.0

Prepared by Walker Monaghan

ENMU-CS 472

2/17/2021

## Contents

<b>1. Introduction .....</b>	<b>2</b>
1.1 Purpose .....	2
1.2 Scope .....	2
1.3 Intended Audience and Reading Suggestions .....	2
1.4 Document Conventions .....	2
1.5 Overview .....	3
<b>2. Overall Description .....</b>	<b>3</b>
2.1 Product Perspective .....	3
2.2 Product Functions .....	4
2.3 User Classes and Characteristics .....	5
2.4 Operating Environment .....	6
2.5 Design and Implementation Constraints .....	6
2.6 User Documentation .....	6
2.7 Assumptions and Dependencies .....	6
<b>3. External Interface Requirements .....</b>	<b>6</b>
3.1 User Interfaces .....	6
3.2 Software Interfaces .....	7
3.3 Communications Interfaces .....	7
<b>4. System Features .....</b>	<b>8</b>
4.1 Customer - Registration .....	8
4.2 Customer - View Upcoming Shows .....	9
4.3 Customer - Select a Show .....	10
4.4 Customer - Seat Select/Unselect .....	10
4.5 Customer - Add Seats to Cart .....	11
4.6 Customer - Manage Cart .....	12
4.7 Customer - Confirm Purchase .....	12
4.8 Customer - Print Tickets .....	13
4.9 Admin - Admin Portal .....	13
4.10 Admin - Add/Remove Shows .....	14
4.11 Admin - Adjust Seat Pricing .....	15
4.12 Admin - Reporting .....	15
<b>5. Other Nonfunctional Requirements .....</b>	<b>16</b>
5.1 Performance Requirements .....	16
5.2 Safety Requirements .....	16
5.3 Security Requirements .....	16
5.4 Software Quality Attributes .....	16
5.5 Business Rules .....	17
<b>6. Other Requirements .....</b>	<b>17</b>
<b>7. Appendix A: Analysis Models .....</b>	<b>17</b>
7.1 Admin.addShow() .....	17
7.2 Customer.buy_ticket() .....	18
7.3 Ticket.update_seat() .....	18
<b>8. Appendix B: Glossary .....</b>	<b>19</b>

## Revision History

Name	Date	Reason For Changes	Version

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to outline in detail the work being undertaken and considered by this team in relation to delivering a comprehensive booking system solution to our client, the Portales Theatre. Within you will find information on our goals, scope, target audience and interactions with this software between both customer and client-administrator users.

### 1.2 Scope

The scope for this project is to provide the requested and required booking/reservation features in an available and secure web environment. Up to and including browsing, review, selection, purchasing, and receipt records for customers as well as product scheduling and reporting for client-administrator users.

### 1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project leads, client-users, testers, and documentation writers.

This document contains comprehensive and evolving knowledge for project stakeholders.

This document should be reviewed top-down for thorough understanding of the project at hand. Follow up references can be best navigated through the Table of Contents above.

### 1.4 Document Conventions

No standards or typographical conventions have been implemented (1.0) as of this current document version. Requirement statements will have clearly indicated priority when reviewed through this document.

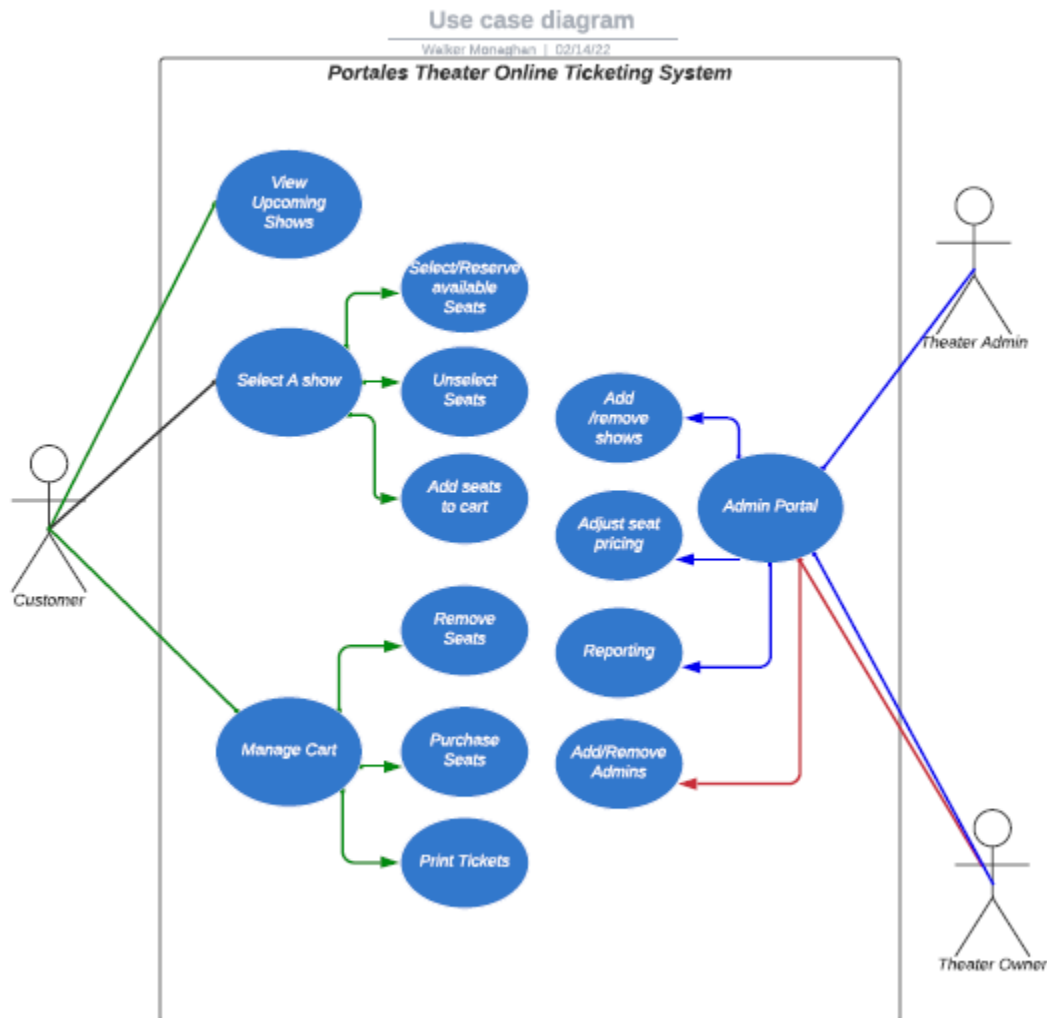
## **1.5 Overview**

The following documents communicates general descriptions and details pertaining to the components built for this project. Section 2 will cover the overall description of this project. Section 3 will provide information on External Interface Requirements such as user interfaces, hardware interfaces, software interfaces, and communications interfaces. Section 4 outlines key features in this system, providing additional detail on each feature as reviewed. Section 5 will cover other nonfunctional requirements. This includes Performance requirements, Safety Requirements, Security Requirements, Software Quality Attributes, and Business Rules. Section 6 serves as a catch-all for any other Requirements not covered in previous sections. The reader will find a series of Appendix's at the end of this document which include a Glossary of terms, Analysis Models, and our Running to Be Determined list.

## **2. Overall Description**

### **2.1 Product Perspective**

This new, self-contained product is a collection of components intended to provide a comprehensive, reliable, and UX optimized solution to our client, the Portales Theatre. As the flagship undertaking of this development team, we have built custom components to deliver upon product requirements.



## 2.2 Product Functions

### • Customer Functions

- Website that allows customers to:
  - Register as a new customer OR login to their account
    - View upcoming showings
    - Select upcoming showing (multiple times)
      - View available/unavailable seats per showing
      - Select seats, with temporary reservation upon action
      - unselect seats, releasing reservation
      - Add selected seats to “Cart”
    - Purchase seats with a credit card
      - Present total
      - Final confirmation page (pre-payment, opportunity to return and make changes)

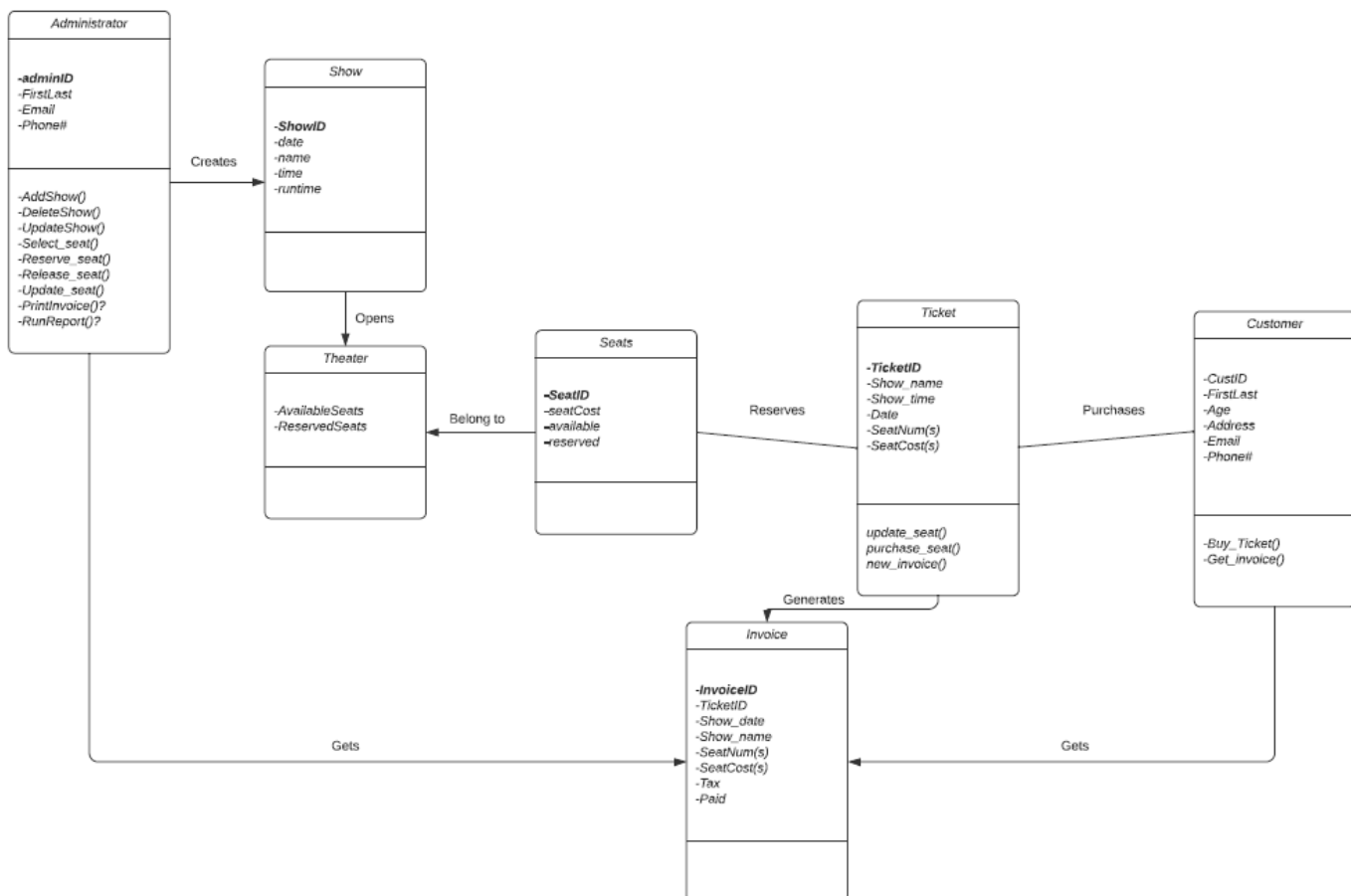
- Print tickets/Invoice
- **Administrator Functions**
  - “Management area” Access Login page
    - Administrator actions page
      - Can Add/Delete plays and their schedules (days and times)
      - Edit price of seating
        - Ability to select seats and adjust pricing of those selected seats
      - Reporting
        - See how many seats have been sold for a specific play

## 2.3 User Classes and Characteristics

### Customer Class – Priority Experience Driver

UML class diagram

Walker Monaghan | 2/27/22



- May view, select, and purchase tickets to showing(s).

### Theater Admin Class

- Access to admin portal

- May add/remove plays
- May edit seat pricing
- May run reports

#### **Theater Owner Class**

- All access/Privilege of Theater Admin class
- May add/remove Admins

## **2.4 Operating Environment**

This software will be web-based and operate on any optimized browser such as Chrome, Safari, Edge, and Firefox.

This software must communicate with a web-based database for operation.

## **2.5 Design and Implementation Constraints**

#### *Considerations:*

- Interaction must not be cumbersome for customers or administrators, timing is important.
- Webpage must be linked to web-hosted database
- Webpage must ensure customer is registered for the session
- Webpage must verify admin is verified

## **2.6 User Documentation**

- A comprehensive user manual will be delivered along with this software

## **2.7 Assumptions and Dependencies**

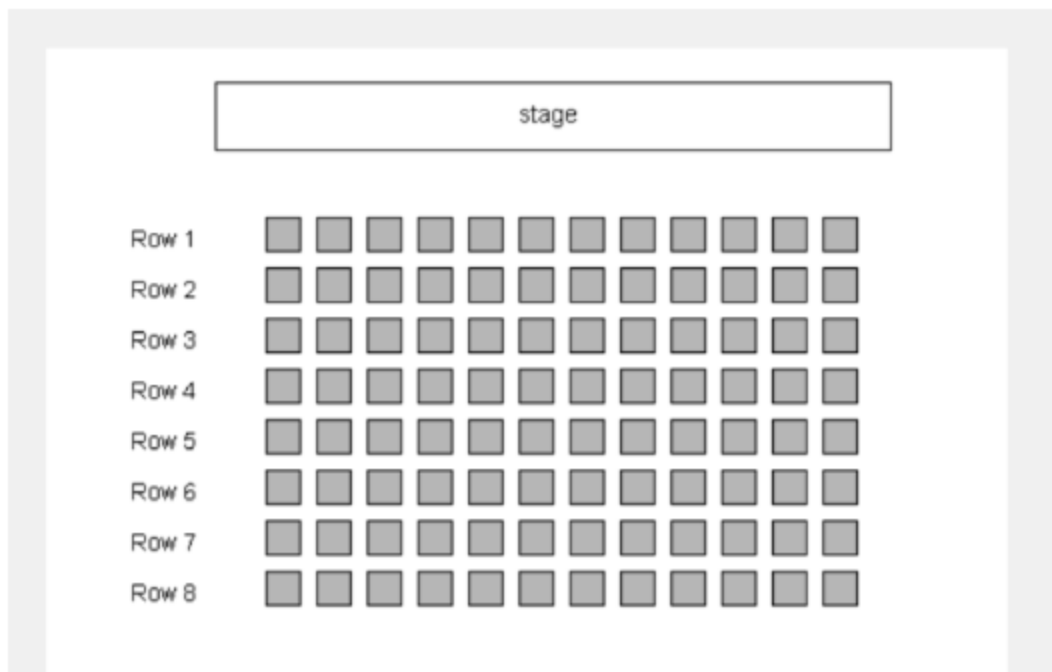
#### *Assumptions*

- There is only one theater, therefore only 1 show at any given time
- Customers do not need an account management portal to access previous purchases in the app, invoices should be emailed to ensure this.

# **3. External Interface Requirements**

## **3.1 User Interfaces**

#### *Example Interface per requirements gathering:*



### 3.2 Software Interfaces

Customer

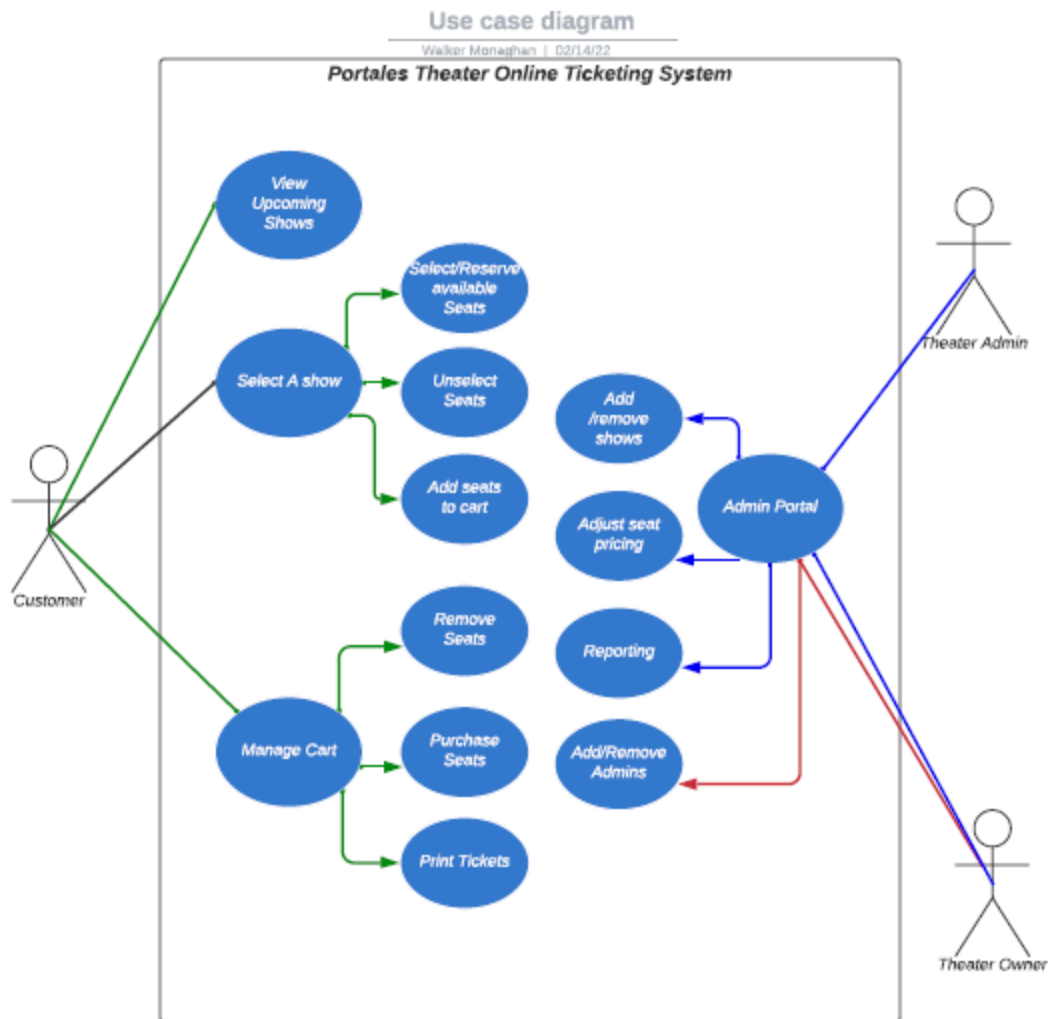
- Register -> CustomersTable
  - View Upcoming Shows -> AllPlaysWithPrices
  - Select A show -> AllPlaysWithPrices
  - seat selection -> StatusofSeatsPlay#
  - Add seats to cart -> StatusofSeatsPlay#
  - Manage Cart ->SeatsPerCustomer
  - Remove Seats -> SeatsPerCustomer
  - Purchase Seats -> SeatsPerCustomer
  - confirm purchase -> SeatsPerCustomer
  - Print Tickets ->SeatsPerCustomer
- Theater Admin -> AdminTable
- Admin Portal -> AllPlaysWithPrices
  - Add/Remove Shows ""
  - Adjust seat pricing ""
  - Admin Reporting "" & StatusofSeatsPlay#

### 3.3 Communications Interfaces

- Electronic Forms
- Email
- Web navigation
- Database communications



## 4. System Features



### 4.1 Customer - Registration

<b>ID:</b>	<b><u>Cust registration</u></b>
<b>Title:</b>	Customer registers into database
<b>Description:</b>	This will allow customers to register in the Portales Theater system, enabling use of the systems seat purchasing. Customer Table:

	customer_id int PRIMARY KEY, first_name varchar(30), last_name varchar(30), address varchar(50), city varchar(50), state varchar(50), zip_code varchar(10), telephone varchar(12), email varchar(50), age int(3)
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Information is not already cached (unique visitor)
<b>Postconditions:</b>	Customer has been added to Customers Table
<b>Main Success Scenario:</b>	The customer will visit the Portales Theater website and be immediately prompted to register before proceeding to view shows and reserve seats.
<b>Extensions:</b>	[Describe all the other scenarios for this use case - including exceptions and error cases.]
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

## 4.2 Customer - View Upcoming Shows

<b>ID:</b>	<b>View_upcoming_shows</b>
<b>Title:</b>	Customer can view all upcoming shows
<b>Description:</b>	This will allow the customer to see all upcoming shows that have been added by admin. Shows where the start time has lapsed will not be visible.
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Static – none
<b>Postconditions:</b>	Static - none
<b>Main Success Scenario:</b>	Customer is able to view an accurate list of upcoming shows to select from
<b>Extensions:</b>	Error – no shows generate Error – lapsed shows generate
<b>Frequency of Use:</b>	High

<b>Status:</b>	Design
----------------	--------

### 4.3 Customer - Select a Show

<b>ID:</b>	<b>Select_show</b>
<b>Title:</b>	Customer selects a show
<b>Description:</b>	A customer is able to select an available upcoming show
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Show must have been visible during view_upcoming_show
<b>Postconditions:</b>	Move to seat_select_unselect
<b>Main Success Scenario:</b>	Customer is able to select an upcoming show and move forward to seat reservations in seat_select_unselect
<b>Extensions:</b>	Error – selects a show that has lapsed Event – selects a show that is sold out
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

### 4.4 Customer - Seat Select/Unselect

<b>ID:</b>	<b>Seat_select_unselect</b>
<b>Title:</b>	Customer selection and unselecting
<b>Description:</b>	“The customer clicks on the available...” (green) “...seats to add them to their list of seats for that play. The user can also click on a previously selected seat to remove it from their list of seats.”
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	A customer will see available green seats and unavailable red seats.
<b>Postconditions:</b>	A customer has selected the quantity and location of seats desired.
<b>Main</b>	A customer, having selected a show, will view available seats and select

<b>Success Scenario:</b>	them for reservation. Upon clicking the seats it should be indicated that that seat (or more) are selected and temporarily reserved within the StatusOfSeatsPlay## table for this show.
<b>Extensions:</b>	Error – No seats available Event – customer selection times out
<b>Frequency of Use:</b>	high
<b>Status:</b>	design

#### 4.5 Customer - Add Seats to Cart

<b>ID:</b>	<b>Add_remove_seat_to_cart</b>
<b>Title:</b>	Add or Remove seats to cart
<b>Description:</b>	This interaction will allow customers to add selected seats to their cart, or, remove selected seats from cart and from reservation
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Desired seats have been selected in seat_select_unselect
<b>Postconditions:</b>	Selected seats are added to cart and a on-screen confirmation is generated that the action is complete. Impacts statusOfSeatsPlay## table and SeatsPerCustomer table
<b>Main Success Scenario:</b>	A customer who has selected and temporarily reserved desired seats is able to “add seats to cart”, generating a confirmation message that the action has been complete. Alternately at any step in the scenario, selected seats that have previously been added to cart can be removed.
<b>Extensions:</b>	Error – no seats have been selected (to add) Error – no seats are in cart (to remove)
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

## 4.6 Customer - Manage Cart

<b>ID:</b>	<b>Manage_cart</b>
<b>Title:</b>	Cart Management
<b>Description:</b>	This module will allow the customer to remove seats that have been added and/or proceed to purchase.
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Customer has had previous interaction that have added seats to the cart.
<b>Postconditions:</b>	Customer may return to add more seats or proceed to payment after reviewing cart.
<b>Main Success Scenario:</b>	A customer who has previously added desired seats to cart may now proceed to edit those cart quantities or proceed to continue to payment.
<b>Extensions:</b>	Event – Cart may be empty, continue shopping
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

## 4.7 Customer - Confirm Purchase

<b>ID:</b>	<b>Confirm_purchase</b>
<b>Title:</b>	confirm purchase
<b>Description:</b>	Screen to accept payment credentials and submit purchase for final processing
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Cart must have had valid items to push through to this function and screen
<b>Postconditions:</b>	Confirmation of payment submitted, optional move to print_tickets
<b>Main Success Scenario:</b>	A customer who had previously confirmed a selection of seats in cart management moved to this function to input payment details and “submit” for final processing.

<b>Extensions:</b>	Error – CC could not be processed Error – Input values invalid (expired or incorrect)
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

#### 4.8 Customer - Print Tickets

<b>ID:</b>	<b>Print_tickets</b>
<b>Title:</b>	Print Tickets
<b>Description:</b>	A customer user is able to print a record of their purchase, showing seat dates, location, show, and price paid.
<b>Primary Actor:</b>	Customer
<b>Preconditions:</b>	Successfully moved past confirm purchase screen
<b>Postconditions:</b>	Files handed off to user print system
<b>Main Success Scenario:</b>	a customer, having successfully purchased tickets through Confirm Purchase is given the option to “Print Tickets” through a button. This will generate a printable pdf of purchased seats from this current transaction.
<b>Extensions:</b>	n/a
<b>Frequency of Use:</b>	High
<b>Status:</b>	Design

#### 4.9 Admin - Admin Portal

<b>ID:</b>	<b>Admin_portal</b>
------------	---------------------

<b>Title:</b>	Admin portal
<b>Description:</b>	Houses exclusive access to a serious of admin resources
<b>Primary Actor:</b>	Admin
<b>Preconditions:</b>	Static
<b>Postconditions:</b>	Admin user, having authenticated, sees a variety of admin-only tools: add/remove shows, adjust seat pricing, run reports.
<b>Main Success Scenario:</b>	Upon initial access, an admin must be authenticated. Following authentication admin tools will be visible.
<b>Extensions:</b>	Access error – authentication failed Access error – Too many failed attempts
<b>Frequency of Use:</b>	Medium
<b>Status:</b>	Design

#### 4.10 Admin - Add/Remove Shows

<b>ID:</b>	<b>Add_remove_shows</b>
<b>Title:</b>	Add/Remove Shows
<b>Description:</b>	An authenticated admin can add additional shows or remove shows from the plays table.
<b>Primary Actor:</b>	Admin
<b>Preconditions:</b>	Authentication
<b>Postconditions:</b>	Modified pays table
<b>Main Success Scenario:</b>	a admin is able to add or remove multiple shows through a form that acts upon the website database. Following this they are able to close out, sign out, or navigate to other admin functions.
<b>Extensions:</b>	Error – missing key values Error – (removal) show not found

<b>Frequency of Use:</b>	Medium
<b>Status:</b>	Design

#### 4.11 Admin - Adjust Seat Pricing

<b>ID:</b>	<b>Adjust_seat_pricing</b>
<b>Title:</b>	Adjust Seat Pricing
<b>Description:</b>	Admin is able to adjust the pricing of seats in individual shows
<b>Primary Actor:</b>	Admin
<b>Preconditions:</b>	A valid show must be selected (a show that has not yet lapsed)
<b>Postconditions:</b>	Seat prices have been modified and updated in the website database, so they are immediately reflected from customer view
<b>Main Success Scenario:</b>	An admin is able to select an upcoming show, select multiple seats in the show, and adjust the pricing of those seats in groups (or individually if they choose).
<b>Extensions:</b>	Consideration: what if seats have already been purchased, are those locked out of pricing changes?  Consideration: if a show is already "published" can changes still be made?
<b>Frequency of Use:</b>	Medium
<b>Status:</b>	Design

#### 4.12 Admin - Reporting

<b>ID:</b>	<b>Admin_reporting</b>
<b>Title:</b>	Admin Reporting
<b>Description:</b>	An administrator is able to run a report on currently sold seats per show
<b>Primary Actor:</b>	Admin
<b>Preconditions:</b>	Admin authenticated, visible way to select a show and click a button to generate a report
<b>Postconditions:</b>	Report is generated with the option to print/export or clear direction to run another report.



<b>Main Success Scenario:</b>	An authenticated admin will access this page and see available filters to narrow down shows in question. Following that selection, a button will be pressed to generate a report showing seats sold details.
<b>Extensions:</b>	Error – no shows meet selected filters
<b>Frequency of Use:</b>	Low
<b>Status:</b>	design

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

### 5.2 Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>*

### 5.3 Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

### 5.4 Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## 5.5 Business Rules

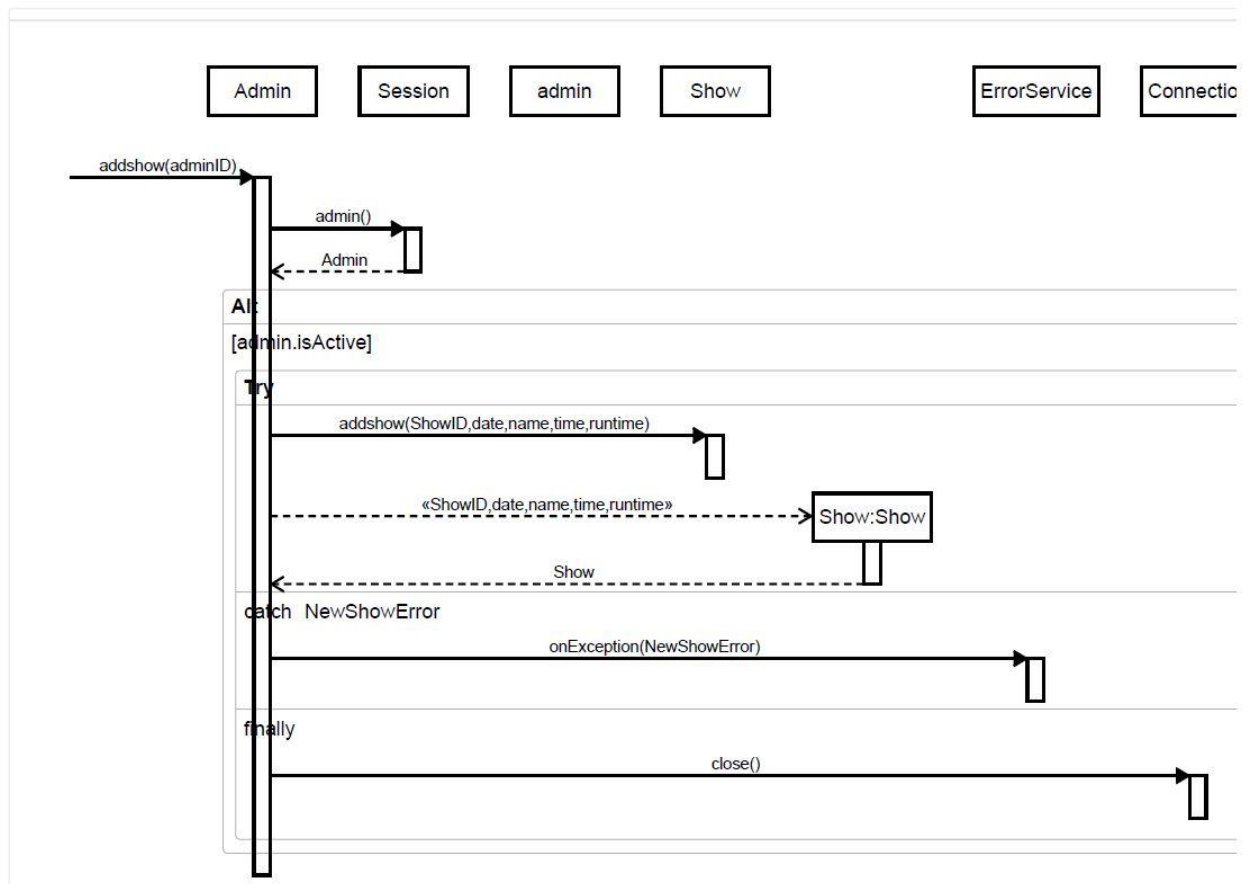
<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

## 6. Other Requirements

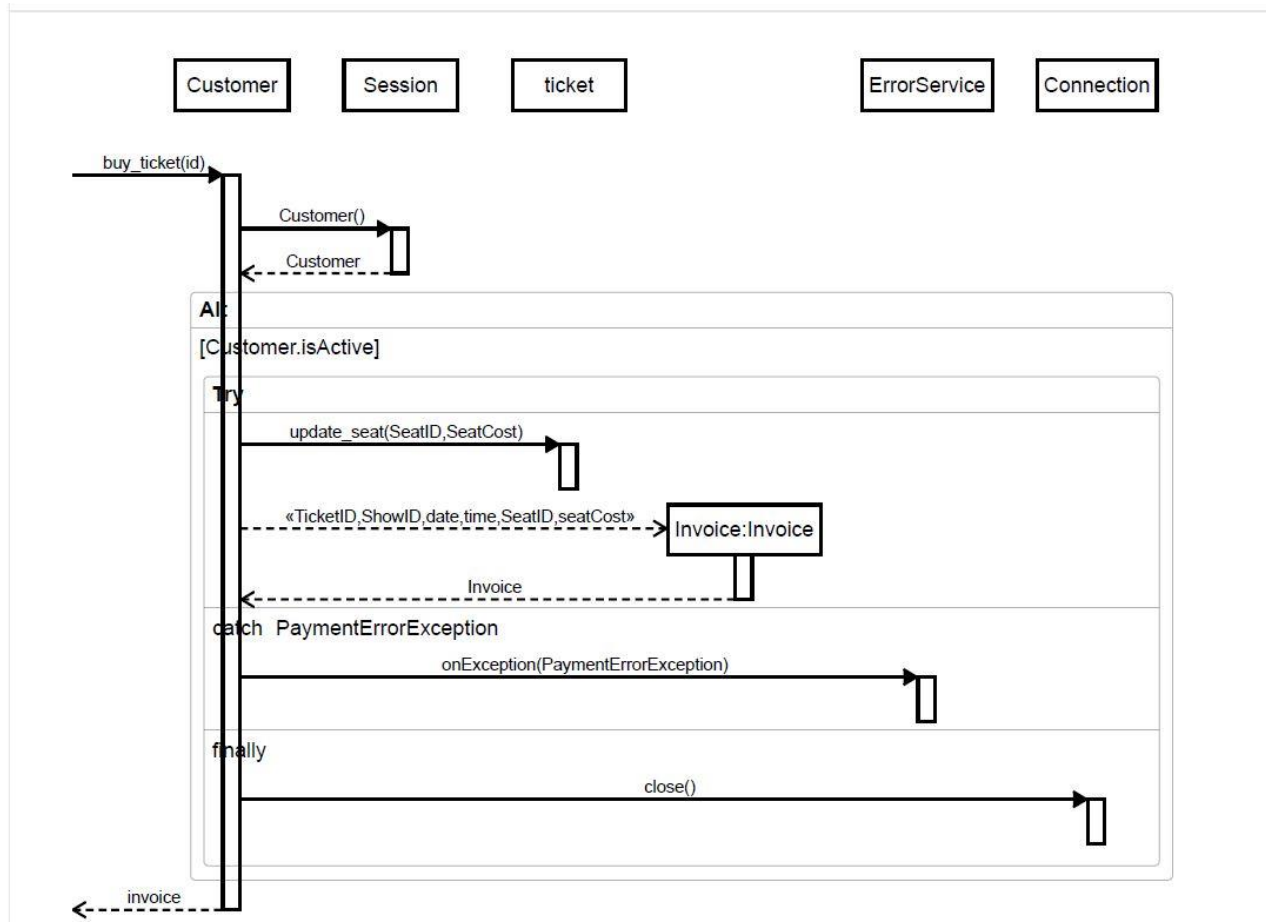
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

## 7. Appendix A: Analysis Models

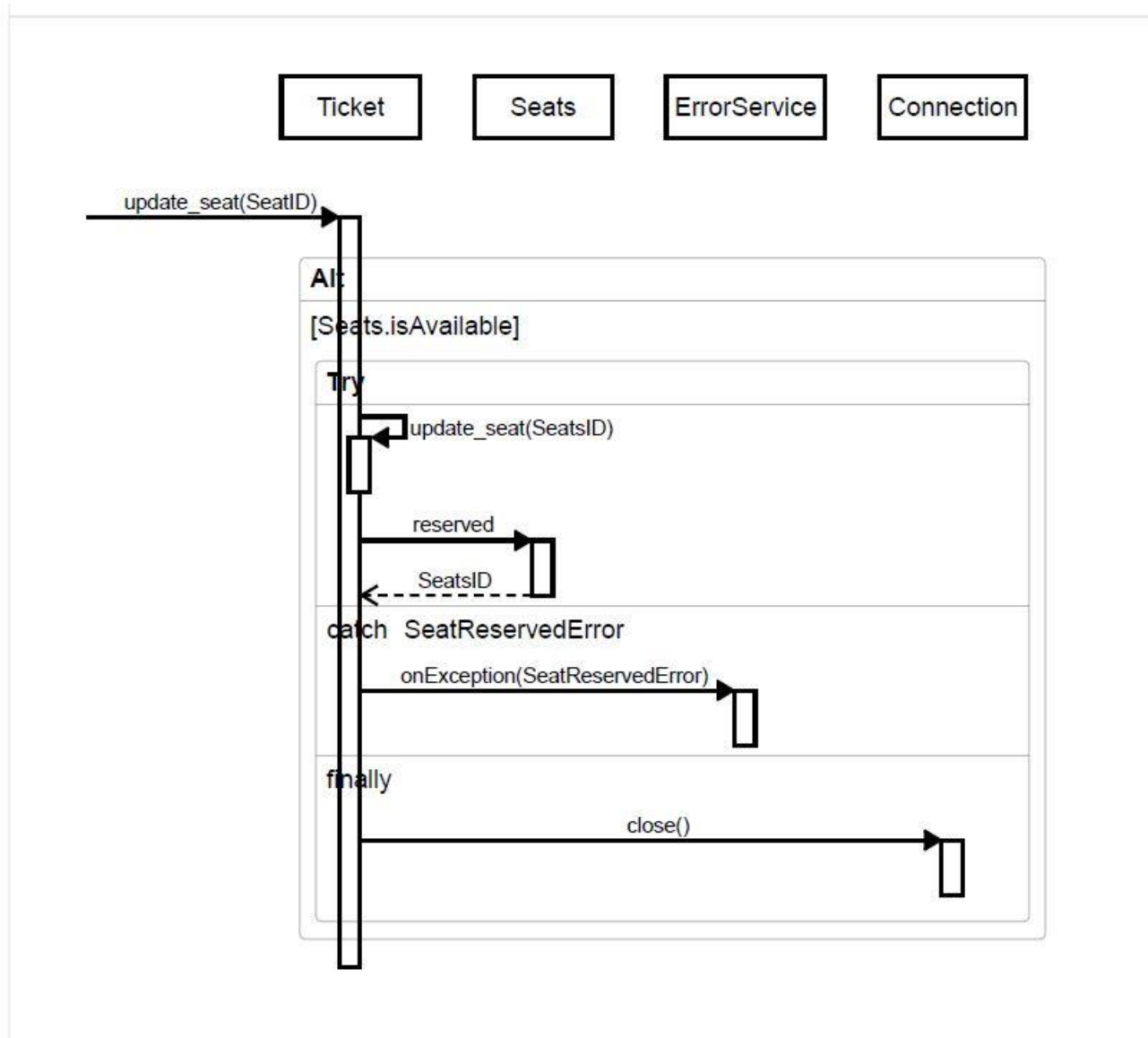
### 7.1 Admin.addShow()



## 7.2 Customer.buy\_ticket()



## 7.3 Ticket.update\_seat()



## 8. Appendix B: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>