

Software Requirements Specification

for

Project-Theater

Version 1.0

Prepared by Walker Monaghan

ENMU-CS 472

2/17/2021

Contents

1. Introduction	2
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
2. Overall Description	3
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
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Software Interfaces.....	7
3.3 Communications Interfaces.....	7
4. System Features.....	8
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 Admin - Admin Portal.....	13
4.9 Admin - Add/Remove Shows	13
4.10 Admin - Adjust Seat Pricing	14
4.11 Admin - Reporting	15
5. Other Nonfunctional Requirements	15
5.1 Performance Requirements	15
5.2 Safety Requirements	15
6. Other Requirements.....	15
7. Appendix A: Analysis Models	16
7.1 Admin.addShow().....	16
7.2 Customer.buy_ticket().....	16
7.3 Ticket.update_seat().....	17
8. Appendix B: Glossary	18

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 conversions have been implemented (1.0) as of this current document version. Requirement statements will have clearly indicated priority when reviewed through this document.

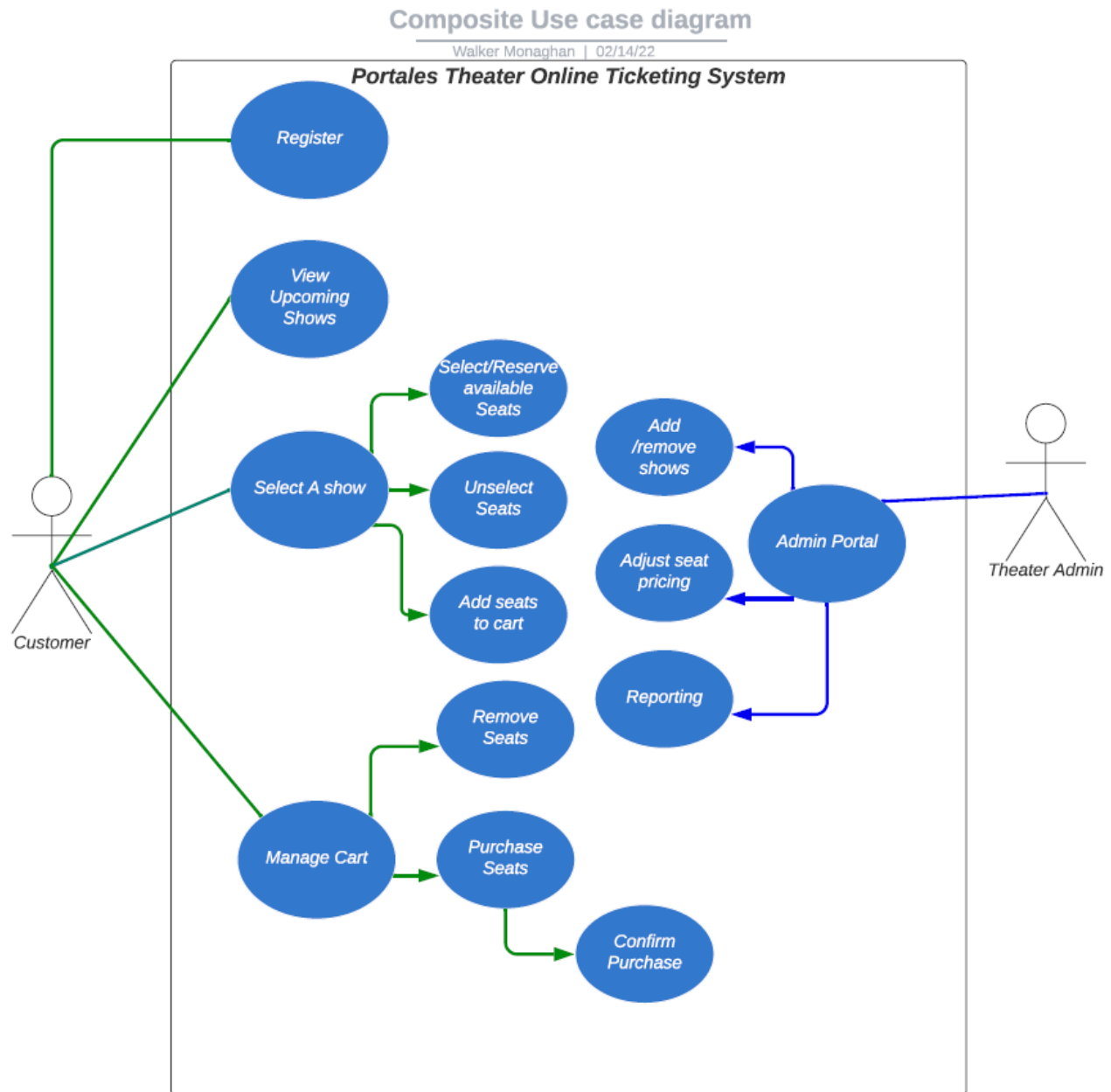
1.5 Overview

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

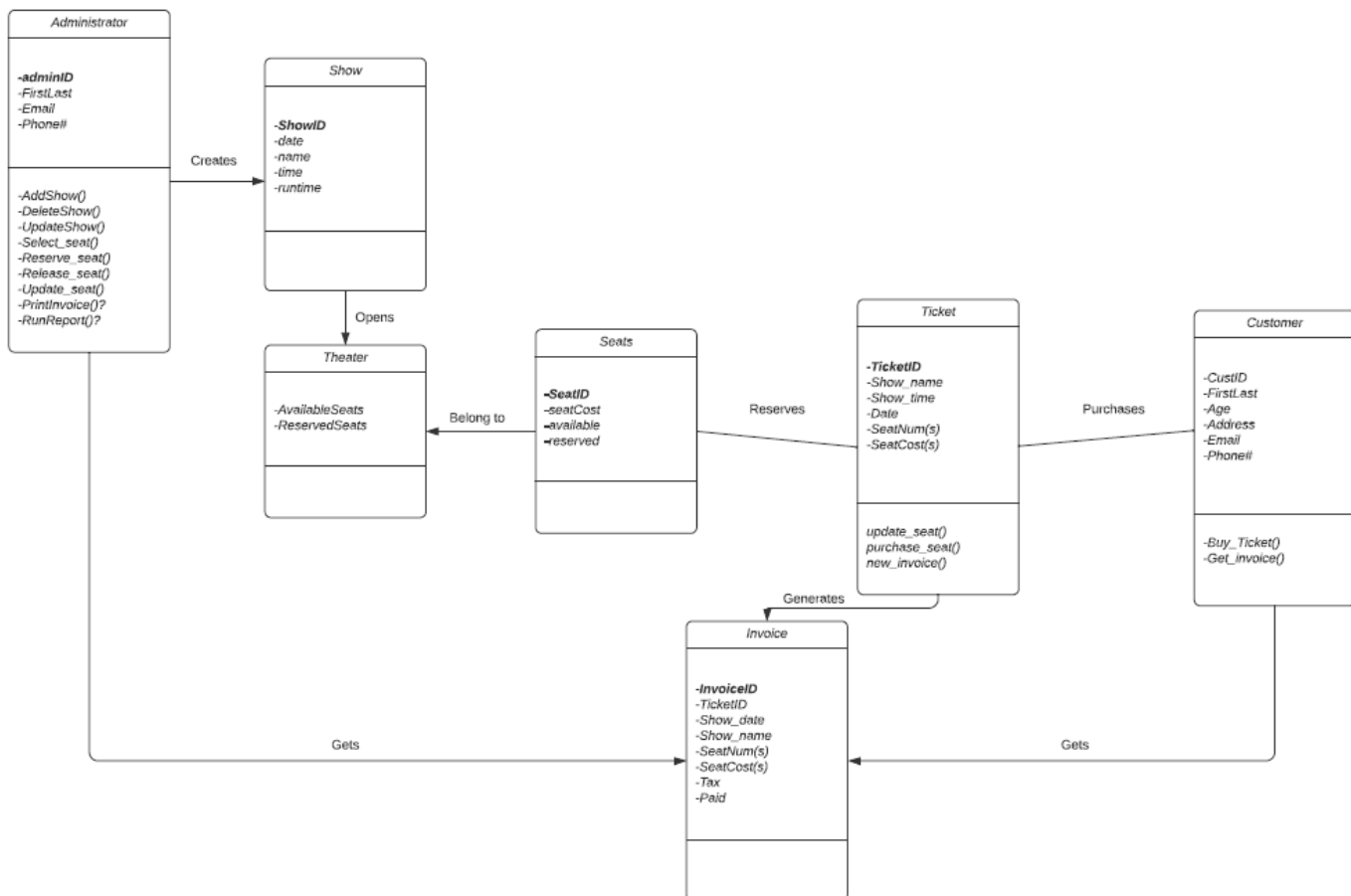
- Add selected seats to “Cart”
- Purchase seats with a credit card
 - Present total
 - Final confirmation page (pre-payment, opportunity to return and make changes)
- **Administrator Functions**
 - “Management area” Access Login page
 - Administrator actions page
 - Can Add/Delete plays and their schedules (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 Managhan | 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

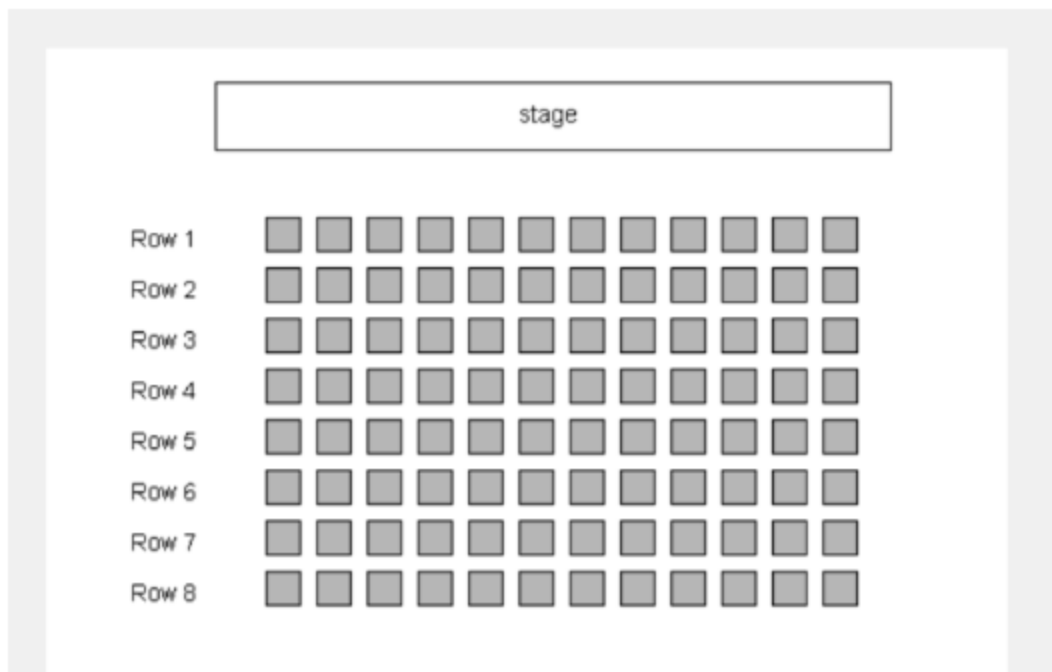
Assumptions

- 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

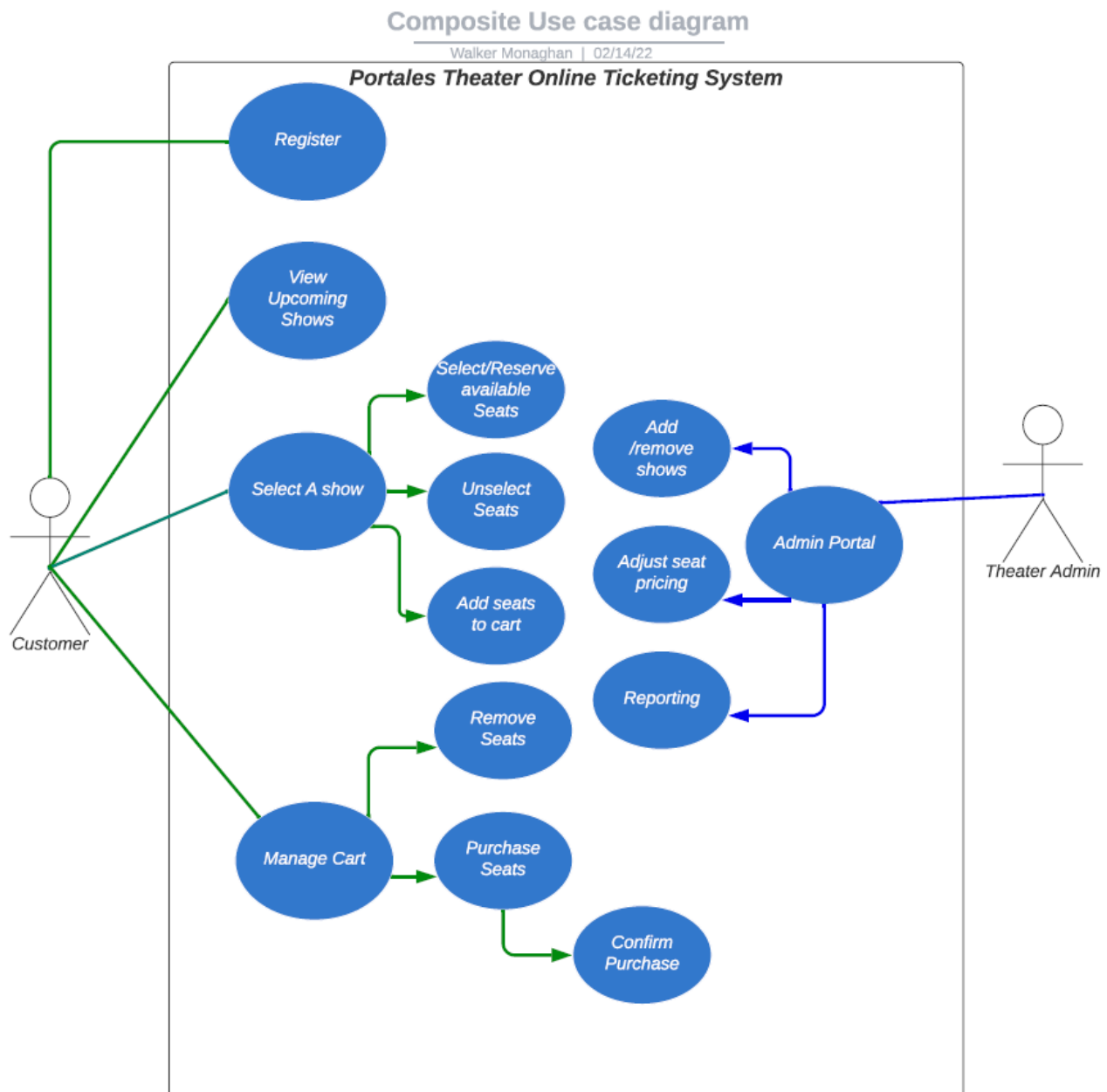
Customer

- 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
- 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

ID:	<u>Cust registration</u>
------------	---------------------------------

Title:	Customer registers into database
Description:	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)
Primary Actor:	Customer
Preconditions:	Information is not already cached (unique visitor)
Postconditions:	Customer has been added to Customers Table
Main Success Scenario:	The customer will visit the Portales Theater website and be immediately prompted to register before proceeding to view shows and reserve seats.
Extensions:	[Describe all the other scenarios for this use case - including exceptions and error cases.]
Frequency of Use:	High
Status:	Design

4.2 Customer - View Upcoming Shows

ID:	View_upcoming_shows
Title:	Customer can view all upcoming shows
Description:	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.
Primary Actor:	Customer
Preconditions:	Static – none
Postconditions:	Static - none
Main Success Scenario:	Customer is able to view an accurate list of upcoming shows to select from
Extensions:	Error – no shows generate

	Error – lapsed shows generate
Frequency of Use:	High
Status:	Design

4.3 Customer - Select a Show

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

4.4 Customer - Seat Select/Unselect

ID:	Seat_select_unselect
Title:	Customer selection and unselecting
Description:	“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.”
Primary Actor:	Customer
Preconditions:	A customer will see available green seats and unavailable red seats.

Postconditions:	A customer has selected the quantity and location of seats desired.
Main Success Scenario:	A customer, having selected a show, will view available seats and select 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.
Extensions:	Error – No seats available Event – customer selection times out
Frequency of Use:	high
Status:	design

4.5 Customer - Add Seats to Cart

ID:	Add_remove_seat_to_cart
Title:	Add or Remove seats to cart
Description:	This interaction will allow customers to add selected seats to their cart, or, remove selected seats from cart and from reservation
Primary Actor:	Customer
Preconditions:	Desired seats have been selected in seat_select_unselect
Postconditions:	Selected seats are added to cart and a on-screen confirmation is generated that the action is complete. Impacts statusOfSeatsPlay## table and SeatsPerCustomer table
Main Success Scenario:	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.
Extensions:	Error – no seats have been selected (to add) Error – no seats are in cart (to remove)
Frequency of Use:	High
Status:	Design

4.6 Customer - Manage Cart

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

4.7 Customer - Confirm Purchase

ID:	Confirm_purchase
Title:	confirm purchase
Description:	Screen to accept payment credentials and submit purchase for final processing
Primary Actor:	Customer
Preconditions:	Cart must have had valid items to push through to this function and screen
Postconditions:	Confirmation of payment submitted, optional move to print_tickets
Main Success Scenario:	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.

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

4.8 Admin - Admin Portal

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

4.9 Admin - Add/Remove Shows

ID:	Add_remove_shows
------------	-------------------------

Title:	Add/Remove Shows
Description:	An authenticated admin can add additional shows or remove shows from the plays table.
Primary Actor:	Admin
Preconditions:	Authentication
Postconditions:	Modified pays table
Main Success Scenario:	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.
Extensions:	Error – missing key values Error – (removal) show not found
Frequency of Use:	Medium
Status:	Design

4.10 Admin - Adjust Seat Pricing

ID:	Adjust_seat_pricing
Title:	Adjust Seat Pricing
Description:	Admin is able to adjust the pricing of seats in individual shows
Primary Actor:	Admin
Preconditions:	A valid show must be selected (a show that has not yet lapsed)
Postconditions:	Seat prices have been modified and updated in the website database, so they are immediately reflected from customer view
Main Success Scenario:	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).
Extensions:	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?
Frequency of Use:	Medium

Status:	Design
----------------	--------

4.11 Admin - Reporting

ID:	Admin_reporting
Title:	Admin Reporting
Description:	An administrator is able to run a report on currently sold seats per show
Primary Actor:	Admin
Preconditions:	Admin authenticated, visible way to select a show and click a button to generate a report
Postconditions:	Report is generated with the option to print/export or clear direction to run another report.
Main Success Scenario:	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.
Extensions:	Error – no shows meet selected filters
Frequency of Use:	Low
Status:	design

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance must not be burdensome for users and optimized interactions between html forms and database interactions is essential to the success of this product.

5.2 Safety Requirements

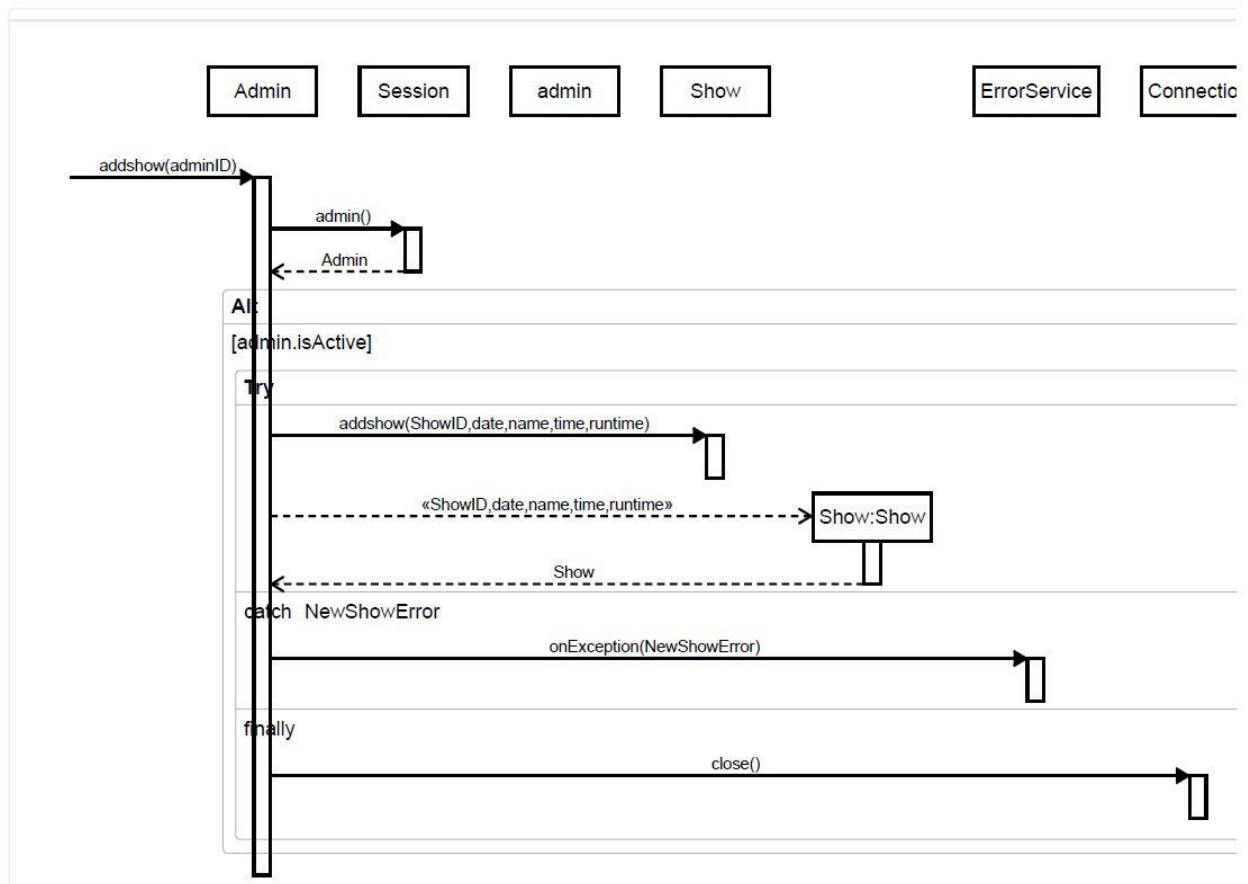
*Admin access must be secure.
 Customer records must be secure
 Customer payment information must be secure
 Website uptime must be a priority*

6. Other Requirements

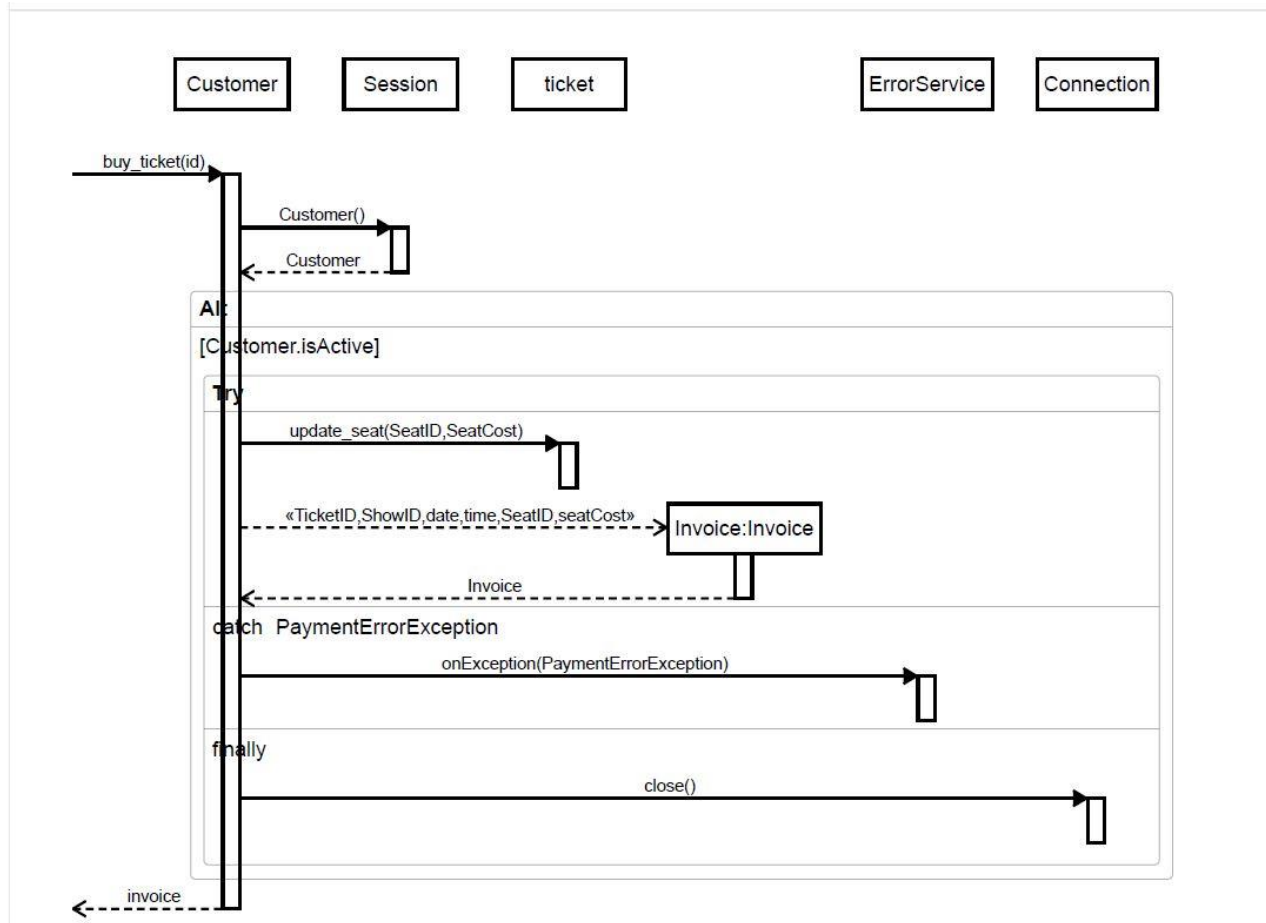
tbd

7. Appendix A: Analysis Models

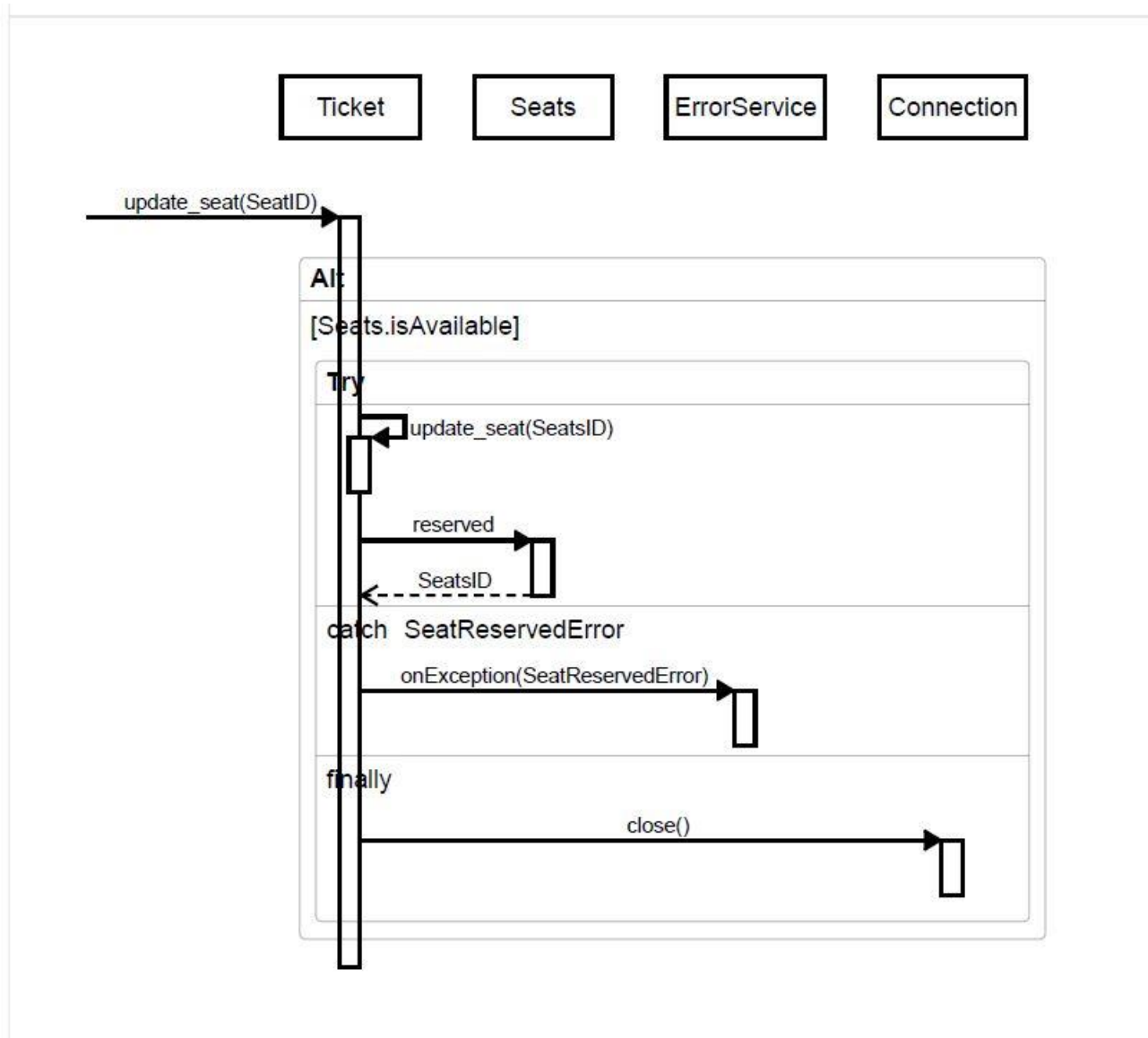
7.1 Admin.addShow()



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.>